UNIT 1 INTRODUCTION TO TCP/IP

Structure **Page Nos**

1.0 Introduction 5

1.1 Objectives 6

1.2 Origin of TCP/IP and Internet 6

1.2.1 Communication 6

1.2.2 Why do we Need the Internet 8

1.2.3 Need of Protocol on Communication 8

1.2.4 Problems in Computer Communication 9

1.2.5 Dealing with Incompatibility 9

1.2.6 A Brief History of the Internet 10

1.2.7 Architecture of the Internet 11

1.3 TCP/IP Layer and Protocols 11

1.4 Network Access Layer 13

1.5 Internet Layer 14

1.5.1 Need for IP Address 14

1.5.2 Classes of IP Address 14

1.5.3 Special Meanings 15

1.5.4 Who Decides the IP Addresses 16

1.5.5 Internet Protocol 16

1.5.6 Address Resolution Protocol (ARP) 18

1.5.7 Reverse Address Resolution Protocol (RARP) 19

1.5.8 Internet Control Message Protocol (ICMP) 19

1.6 Transport Layer 20

1.6.1 Transmission Control Protocol 21

1.6.2 User Datagram Protocol (UDP) 25

1.7 Application Layer 26

1.7.1 Electronic Mail 26

1.7.2 Domain Name System (DNS) 30

1.7.3 How does the DNS Server Works? 32

1.7.4 Simple Network Management Protocol (SNMP) 34

1.7.5 Remote Login: TELNET 36

1.7.6 World Wide Web: HTTP 38

1.8 Networking Example 40

1.9 Summary 42

1.10 Answers to Check Your Progress 43

1.11 Further Readings 44

1.0 INTRODUCTION

Transmission Control Protocol (TCP)/Internet Protocol (IP) is a set of protocols developed to allow computers of all sizes from different vendors, running different operating systems, to communicate or to share resources across a network. A packet switching network research project was started by the USA Government in the late 1960s in 1990s, became the most widely used form of computer networking. This project centered around ARPANET. ARPANET is best-known TCP/IP network.

TCP/IP is the principal UNIX networking protocol and was designed to provide a reliable end-to-end byte stream over an unreliable internetwork. TCP is a connection-oriented protocol while IP is a connectionless protocol. TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual-circuit that two processes can use to communicate. IP provides a connectionless and unreliable delivery system and transfer each datagram independently in the network.

UDP is a connectionless and unreliable protocol running over IP. It adds a checksum to IP for the contents of the datagram and pass members. In this unit we are going to discuss all the protocols of TCP/IP in brief.

1.1 OBJECTIVES

After going through this unit, you should be able to know:

- the need of Communication and Internetworking;
- how the Internet and TCP/IP came into existence;
- the meaning of the terms protocol and standard;
- the architecture of the TCP/IP Protocol Suite;
- difference between the OSI model and the TCP/IP Suite;
- the need and functionalities of Interface layer;
- the need and functionalities of Internet layer;
- the need and functionalities of Transport layer;
- the need and functionalities of Application layer, and
- how communication takes place on the Internet.

1.2 ORIGIN OF TCP/IP AND INTERNET

Before going through the origin of the Internet, let us examine what is communication.

1.2.1 Communication

Communication the process of sharing ideas, information, and messages with others at a particular time and place. Communication is a vital part of personal life and is also important in business, education, and any other situation where people encounter each other. Communication between two people is an outgrowth of methods developed over centuries of expression. Gestures, the development of language, and the necessity to engage in joint action all play a part. Communication, as we see it today, has evolved a long way. We will discuss the primitive modes of communication briefly.

i) Early Methods

Early societies developed systems for sending simple messages or signals that could be seen or heard over a short distance, such as drumbeats, fire and smoke signals, or lantern beacons. Messages were attached to the legs of carrier pigeons that were released to fly home (this system was used until World War I, which started in 1914). Semaphore systems (visual codes) of flags or flashing lights were employed to send messages over relatively short but difficult-to-cross distances, such as from hilltop to hilltop, or between ships at sea.

ii) Postal Services

The postal system is a system by which written documents normally enclosed in envelopes, and also small packages containing other matter, are delivered to

destinations around the world. Anything sent through the postal system is called post. In India the East India Company in Mumbai, Chennai and Calcutta introduced the postal system in 1766, further these postal service became available to the general public. Even after implementing different electronic communication mediums, postal system is still one of the popular communication systems available.

iii) Telegraph

The first truly electronic medium for communication was the telegraph, which sent and received electrical signals over long-distance wires. The first practical commercial systems were developed by the physicist, Sir Charles Wheatstone and the inventor Sir William F. Cooke in Great Britain, and by the artist and inventor Samuel F. B. Morse in the United States. Morse demonstrated the first telegraph system in New York in 1837. But regular telegraph service, relaying Morse code (system of code using on and off signals), was not established until 1844. Telegraphers would translate the letters of the alphabet into Morse code, tapping on an electrical switch, or key. The telegrapher at the other end of the line would decode the tapping as it came in, write down the message, and send it to the recipient by messenger. The telegraph made it possible for many companies to conduct their business globally for the first time.

iv) Telephone

Early devices capable of transmitting sound vibrations and even human speech appeared in the 1850s and 1860s. The first person to patent and effectively commercialise an electric telephone was Scottish-born American inventor Alexander Graham Bell. Originally, Bell thought that the telephone would be used to transmit musical concerts, lectures, or sermons.

The telephone network has also provided the electronic network for new computer-based systems like the Internet facsimile transmissions, and the World Wide Web. The memory and data-processing power of individual computers can be linked together to exchange the data transmitted over telephone line, by connecting computers to the telephone network through devices called modems (modulator-demodulators).

v) Computers and Internet

The earliest computers were machines built to make repetitive numerical calculations that had previously been done by hand. While computers continued to improve, they were used primarily for mathematical and scientific calculations, and for encoding and decoding messages. Computer technology was finally applied to printed communication in the 1970s when the first word processors were created.

At the same time computers were becoming faster, more-powerful and smaller, and networks were developed for interconnecting computers. In the 1960's the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defence, along with researchers working on military projects at research centres and universities across the country, developed a network called the ARPANET, for sharing data and processing time of uniform computer connection over specially equipped telephone lines and satellite links. The network was designed to survive the attack or destruction of some of its parts and continue to work.

Soon, however, scientists using the ARPANET realised that they could send and receive messages as well as data and programs over the network. The ARPANET became the first major electronic-mail network; soon thousands of researchers all over the world used it. Later on the National Science Foundation (NSF) helped connect more universities and non-military research sites to the ARPANET, and renamed it the Internet because it was a network of networks among many different organisations.

Today, the Internet is the widely known computer network. It uses interconnection of computer system by both wired and wireless. Smaller networks of computers, called Local Area Networks (LANs), can be installed, in a single building or for a whole organisation. Wide Area Networks (WANs) can be used to span a large geographical area. LANs and WANs use telephone lines, computer cables, and microwave and laser beams to carry digital information around a smaller area, such as a single college campus. Internet can carry any digital signals, including video images, sounds, graphics, animations, and text, therefore it has became very popular communication tool.

1.2.2 Why do we Need the Internet?

The main reason is that each computer network is designed with a specific purpose. For example, LAN is used to connect computers in a smaller area, and it provides fast communication. As a result, networks become specialised identify. In many cases, these networks do not use the same hardware and software technology. It means that, a computer can communicate with the computers attached to the same network, because they are intercompatible. As more and more organisations had multiple computer networks, this became a major issue. As a result, the concept of internetworking (internet) came into being. This means that there should be a network of all physically separate networks.

1.2.3 Need of Protocol on Communication

All methods of communication described above follow a protocol. Protocol is nothing but a convention. To signify that "Everything is ok and the train can start by a green flag is also a protocol. When we write a letter we follow a protocol. If we look at them carefully, we will find that protocols normally have hidden layers. A good example is human conversation over the phone which can be used as an analogy for communication using computers.

Assume that X and Y, want to have conversation over the telephone about a cricket match. We call this an *idea*. Assume that each person is taking down what other has to say. Thus, the conversation takes place in the form of several messages. A message is a block of sentence. It could also consist of one world such as OK, yes denoting a positive ***acknowledgement*** (ACK). It could also mean a **negative acknowledgement** (NAK) or a request to repeat such as come again, pardon me. All this happens both ways.

At the level of idea, X and Y feel that they are discussing a cricket match. However, in reality, the conversation consists of a number of messages.

A message could be too long. It may not be wise for X to speak for half an hour, only to receive a request to repeat. It is therefore necessary to send/receive acknowledgements after each sentence like 'ok', 'come again' etc. A sentence is analogous to a packet in computer world. The sender X will not speak until s/he hears some form of acknowledgement, or will repeat the sentence if s/he receives a negative acknowledgement. An alternative is *timeout* strategy. The speaker speaks a sentence and waits for some time for any acknowledgement. If s/he does not hear anything, s/he repeats the sentence.

Apart from this ***error control,*** we take care of ***flow control***. Flow control refers to the speed mismatch between the listener and speaker. If the speaker speaks too fast, the listener will say go-slow. In computer world, if the receiving computer is not fast enough, and cannot hold any more data, it requests the sender to wait or control the transfer by slowdown.

Therefore, in computer communication, both speaker and lister should agree on the communication language/syntax, scheme of acknowledgement, during flow control, machine error control mechanism, etc. Thus, we can say that the conversation is

governed by some set of rules known to both the parties. This set of rules is called protocol and it necessary for disciplined manner of conversation/communication.

1.2.4 Problems in Computer Communication

The same concept of protocols described above applies to computer communication. The main difference is that earlier, the devices (telephones) on both sides are compatible. However, in an Internet work, it may or may not be so. The concept of Internetworking though, highly desirable, is not easily achievable.

Thus, any two networks, cannot directly communicate by connecting a wire between the networks. For example, one network could represent a binary 0 by–5 volts, another by +5 volts. Similarly, one could use a packet size of 128 bytes, whereas other could use 256 byte packets. The method of acknowledgement or error detection could be different. There could be many such differences.

1.2.5 Dealing with Incompatibility

The incompatibility issues are handled at two levels:

i) Hardware Issues

At the hardware level, an additional component called router is used to connect physically distinct networks as shown in *Figure 1*. A router connects to the network in the same way as any other computer. Any computer connected to the network has a Network Interface Card (NIC), which has the address (network id+host id), hard coded into it. A router is a device with more than one NICs. Router can connect incompatible networks as it has the necessary hardware (NIC) and protocols (TCP/IP).

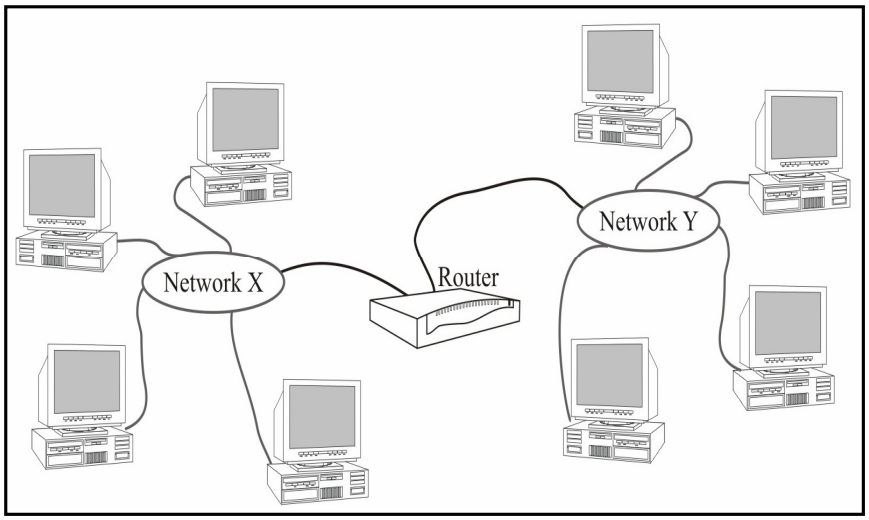


Figure 1: Router connects the Networks

ii) Software Issues

The routers must agree about the way information would be transmitted to the destination computer on a different network, since the information is likely to travel through different routers, there must be a predefined standard to which routers must confirm. Packet formats and addressing mechanism used by the networks may differ. One approach could be to perform conversion and reconversion corresponding to different networks. But this approach is difficult and cumbersome. Therefore, the Internet communication follows one protocol, the TCP/IP protocol suite. The basic idea is that it defines a packet size, routing algorithms, error control, flow control methods universally.

It would be unwise to club all these features in a single piece of software ─ it would make it very bulky. Therefore, all these features are logically sub-grouped and then the sub-groups are further grouped into groups called layers. Each layer has an interface with the adjacent layers, and performs specific functions.

Need for Layering

Since it is difficult to deal with complex set of rules, and functions required for computer networking, these rules and functions are divided with logical groups called layers. Each layer can be implemented interdependently with an interface to other layers providing with services to it or taking its services like flow control and error control functions are grouped together and the layer is called data link layer. Speech in telephone conversation is translated, with electrical segments and vice-versa. Similarly in computer system the data or pattern are converted into signals before transmitting and receiving. These function and rules are grouped together in a layer called physical layer.

1.2.6 A brief history of the Internet

Internet is made up of thousand and thousands of interconnected networks. Although it has become extremely popular in the last decade, it came into being only in 1969.

ARPANET

In the mid 1960s, the Advanced Research Projects Agency (ARPA) in the US Department of Defence (DoD) wanted to find a way to connect computers so that their funded researchers could share their findings. In 1967, ARPA proposed its idea for ARPANET, a small network for connecting computers.

Birth of Internet and TCP/IP

In 1972, Vint Cerf and Bob Kahn, who were part of core ARPANET group, started Internetting Project with the aim to connect different networks together. Diverse interfaces, different packet sizes, diverse transmission rates presented many difficulties. Kahn Cerf proposed the idea of gateway as an intermediate hardware to transfer data from one network to another.

In 1973, they presented a landmark paper outlining the protocols for end-to-end communication. This paper on TCP/IP included concepts like datagram, gateway, and encapsulation. At this time, responsibility of ARPANET was handed to Defence Communication Agency (DCA).

Came 1977 and communication between networks was made possible. Internet consisting of three different networks, namely, ARPANET, packet radio, and packet satellite was now possible.

In 1978 noticing the popularity of BSD (Berkley Software Distribution) UNIX, which included network capabilities, ARPA signed a contract with Berkley under which TCP/IP software was incorporated in the operating system itself.

In 1983 original ARPANET protocols were abolished and TCP/IP was made de facto standard for Internet.

Time Line

- 1969 Four-node ARPA established
- 1972 Internetting Project begins
- 1973 Development of TCP/IP suite begins

- 1977 An Internet tested using TCP/IP
- 1978 UNIX distributed to universities
- 1983 TCP/IP became the official protocol for ARPANET.

1.2.7 Architecture of the Internet

Internet is organised to form a hierarchy. At the top, there is a very high-speed backbone and at the other end, there are users. There are Network Access Providers (NAP) and Internet Service Providers (ISP) at the intermediate level as shown in the *Figure 2.*

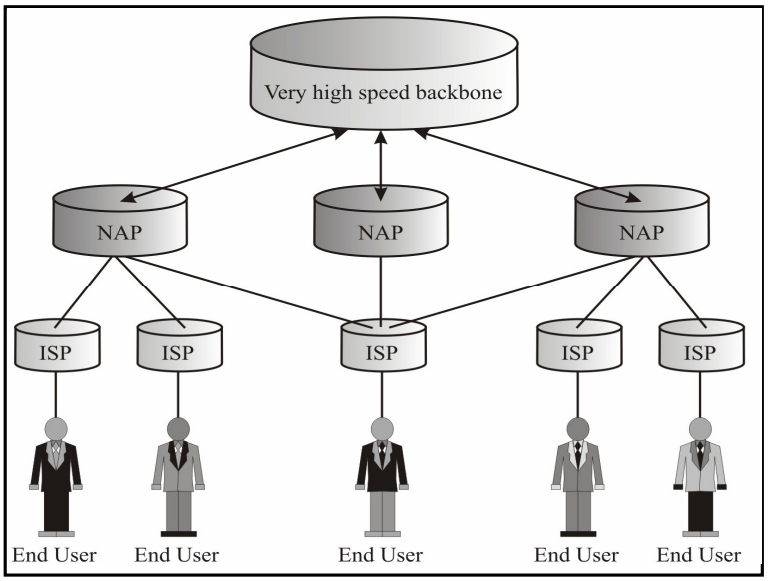


Figure 2: Architecture of Internet

A home user dials into the ISP, may be using a twisted pair telephone connection using a modem. The ISP connects to one of the Network Access Providers, which in turn, connects to the high-speed backbone at a Network Access Point. Network Access Point serves the purpose of connecting backbone networks to provide connectivity between end users.

1.3 TCP/IP LAYERS AND PROTOCOLS

The TCP/IP model is made up of four layers: interface layer, network, transport, and application. The first layer of TCP/IP (Application layer) is similar to the first three layers (Application, presentation and Session layer) of the OSI model. The services of transport layers of both the models are similar. Further, the services of network layers in both models are also similar, while some time network layer is also known as Internet layer. The last layer of TCP/IP is interface layer, which includes the services of data link layer and physical layer of OSI model. In OSI model, each layer takes the services of the lower layer. Whereas the layers of TCP/IP protocol suite contain relatively independent protocols.

The mapping of OSI & TCP/IP model is shown in the *Figure 3*, it also shows different protocols included in the TCP/IP model.

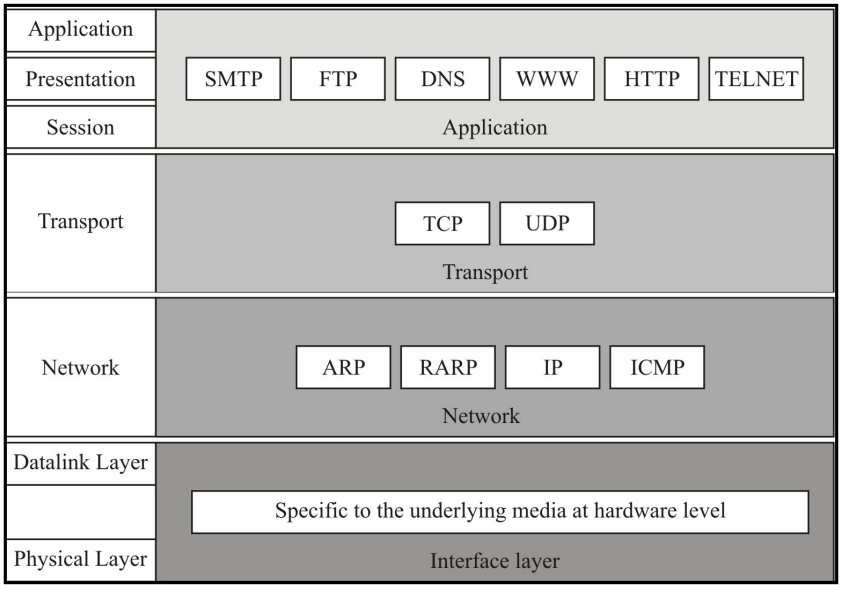


Figure 3: Mapping of OSI and TCP/IP model

Layers of TCP/IP Protocol Suite

As we know TCP/IP contains four layers and each layer has its specific functions, in the following section lets find out the functions of each layer of TCP/IP.

Interface layer or (Physical + Data Link Layer)

The physical layer deals with the hardware level like, transmission media, connections and the voltage for digital signals. The data link layer deals with media access and control strategies, frame format etc.

Internet Layer or Network Layer

The Internet layer is an important layer in the protocol suite. At this layer, TCP/IP supports Internetworking Protocol (IP). IP is a host-to-host protocol. This layer is responsible for the format of datagrams as defined by IP, and routing a datagram or packet to the next hop, but is not responsible for the accurate and timely delivery of datagrams to the destination in proper sequence. IP allows raw transmission functions allowing user to add functionalities necessary for given application. Ensuring maximum efficiency, TCP/IP supports four other protocols: ARP, RARP, ICMP and IGMP in this layer.

- **Address Resolution Protocol (ARP)**

  On a LAN, each machine is identified with a unique physical address imprinted on the network interface card. ARP is used to find the physical address of a machine when its IP address is known.

- **Reverse Address Resolution Protocol (RARP)**

  It is used to find the IP address of a machine when its physical address is known. It is used when a diskless computer is booted or a computer is connected to the network for the first time.

- **Internet Control Message Protocol (ICMP)**

  IP is unreliable are best effort delivery. In case of failures ICMP is used to send notifications to the sender about packet problems. It sends error and query messages.

- **Internet Group Message Protocol (IGMP)**

  It is used for multicasting, which is transmission of a single message to a group of recipients.

Transport Layer

At this layer, TCP/IP supports two protocols: TCP, UDP, IP is host-to-host protocol, which can deliver the packet from one physical device to another physical device. TCP, UDP, are transport level protocols, responsible for delivering a packet from one process on a device to another process on the other device.

User Datagram Protocol (UDP)

It is simpler of the two protocols. It does not provide reliability. It is, therefore faster, and using for applications in which delay is intolerable (in case of audio and video).

Transmission Control Protocol (TCP)

TCP is reliable, connection oriented protocol. By connection oriented, we mean that a connection must be established between both ends before either can transmit data. It ensures that communication is error-free and in sequence.

Application Layer

As said earlier, it is closer to combined session, presentation, and application layer of OSI model. It allows the user to run various applications on Internet. These applications are File Transfer Protocol (FTP), remote login (TELNET), email (SMTP), WWW (HTTP). The session layer of OSI model is almost dropped in TCP/IP.

☞ Check Your Progress 1

1) How is the TCP/IP model different from OSI model?

……..……

……..

……..……

……..……

2) What are the different layers of TCP/IP protocol suite? Explain.

……..……..

……..……

……..……

1.4 NETWORK ACCESS LAYER

The design of TCP/IP hides the function of this layer from users—it is concerned with getting data across a specific type of physical network (such as Ethernet, Token Ring, etc.). This design reduces the need to rewrite higher levels of a TCP/IP stack when new physical network technologies are introduced (such as ATM and Frame Relay).

The functions performed at this level include encapsulating the IP datagrams into *frames* that are transmitted by the network. It also maps the IP addresses to the physical addresses used by the network. One of the strengths of TCP/IP is its addressing scheme, which uniquely identifies every computer on the network. This IP address must be converted into whatever address is appropriate for the physical network over which the datagram is transmitted.

1.5 INTERNET LAYER

The Internet layer is an important layer in the protocol suite. At this layer, TCP/IP supports Internetworking Protocol (IP). IP is host-to-host protocol. This layer is responsible for the format of datagrams as defined by IP, and routing and forwarding a datagram or packet to the next hop, but is not responsible for the accurate and timely delivery of datagrams to the destination in proper sequence. IP allows raw transmission functions allowing user to add functionalities necessary for given application, ensuring maximum efficiency.

1.5.1 Need for IP Address

The primary goal of the Internet is to provide an abstract view of the complexities involved in it. Internet must appear as single network of computers. At the same time network administrators or users must be free to choose hardware or various internetworking technologies like Ethernet, Token ring etc. Different networking technologies have different physical addressing mechanisms. Therefore, identifying a computer on Internet is a challenge. To have uniform addressing for computers over the Internet, IP software defines an IP address which is a logical address. Now, when a computer wants to communicate to another computer on the Internet, it can use logical address and is not bothered with the physical address of the destination and hence the format and size of data packet.

1.5.2 Classes of IP Address

Internet addresses are 32 bits long, written as four bytes separated by periods (full stops). They can range from 0.0. 0.0 to 223. 255. 255. 255. It's worth noting that IP addresses are stored in big-endian format, with the most significant byte first, read left to right. This contrasts with the little-endian format used on Intel- based systems for storing 32- bit numbers. This minor point can cause a lot of trouble for PC programmers and others working with raw IP data if they forget.

IP addresses comprise two parts, the network ID and the host ID. An IP address can identify a network (if the host part is all zero) or an individual host. The dividing line between the network ID and the host ID is not constant. Instead, IP addresses are split into five classes, which allow for a small number of very large networks, a medium number of medium- sized networks and a large number of small networks. The classes of IP address are briefly explained below, the structure of these classes are also shown in *Figure 4*.

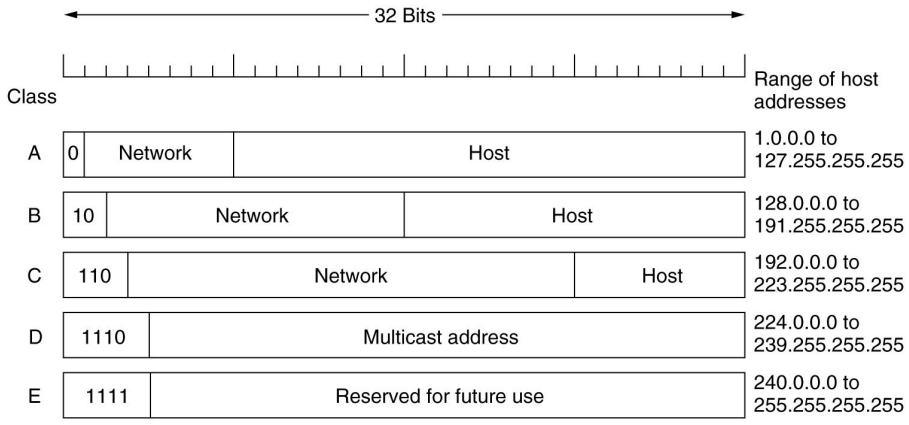


Figure 4: Classes of IP address

Class A addresses have a first byte in the range 0 to 127. The remaining three bytes can be used for unique host addresses. This allows for 126 networks each up to 16 million hosts.

Class B addresses can be distinguished by first byte values in the range 128 to 191 in these addresses, the first two bytes are used for the net ID, and the last two for the host ID, giving addresses for 16,000 networks, each with up to 65536 hosts.

Class C addresses have a first byte in the range 192 to 223. Here, the first three bytes identify the network, leaving just one byte for the individual hosts. This provides for 2 million networks of up to 256 hosts each.

Class D addresses have a first byte in the range 224 to 239. It is designed fo multicasting.

Class E addresses have a first byte in the range 240 to 255.It is reserved for future purposes. The concept is shown here.

Although these addresses make it possible to uniquely identify quite a lot of networks and hosts, the number is not that large in relation to the current rate of expansion of the Internet. Consequently, a new addressing system has been devised which is a part of Internet Protocol version 6 (IPv6). IPv6 won't come into use for a couple of years, and understanding it isn't essential to understanding how IP works in general.

IP addresses can be further divided to obtain a subnet ID. The main net ID identifies a network of networks. The subnet ID lets you address a specific network within that network. This system of addressing more accurately reflects how real- world large networks re-connected together.

You decide how the subnet ID is arrived at by defining a 32- bit value called the subnet mask. This is logically ANDed with the IP address to obtain the subnet address. For example, if a subnet mask was 255.255. 255.0 and an IP address was 128. 124. 14.5, 128. 124 would identify the Class B network, 128. 124. 14 would identify the sub network, and 5 would identify the host on that sub network.

1.5.3 Special Meanings

A few IP addresses have special meanings. A network ID of 0 in an address means "this network", so for local communication only the host ID need be specified. A host ID of 0 means "this host".

A network ID of 127 denotes the loopback interface, which is another way of specifying "this host". The host ID part of the address can be anything in this case, though the address 127. 0.0. 1 is normally used. Packets sent to the loopback address will never appear on the network. It can be used by TCP/ IP applications that run on the same machine and want to communicate with one another.

Addresses in the range 224. x. x. x to 239. x. x. x are Class D addresses, which are used for multi- casting. Addresses 240.x. x. x to 247. x. x. x are reserved for experimental purposes.

Net, subnet and host ID's of all binary ones (byte value 255) are used when an IP packet is to be broadcast. Mercifully, an address of 255. 255. 255.255 does not result in a broadcast to the entire Internet.

Three sets of addresses are reserved for private address space - networks of computers that do not need to be addressed from the Internet. There is one class A address (10.x. x. x), sixteen class B addresses (172. 16. x. x to 172. 31. x. x), and 256 class C addresses (192. 168. 0. x to 192.168. 255. x). If you have equipment which uses IP addresses that have not been allocated by InterNIC then the addresses used should be within one of these ranges, as an extra precaution in case router misconfiguration allows packets to "leak" onto the Internet.

1.5.4 Who Decides the IP Addresses?

IP address of two Computers over the Internet is never same. To ensure this, there is a central authority that issues the IP address. An organisation or individual wanting to connect to the Internet needs to contact local ISP for obtaining a unique IP address at the Metadata the global level, Internet Assigned Number Authority (IANA) allocates a net id to the ISP.

1.5.5 Internet Protocol

IP is the transmission mechanism used by TCP/IP protocols for host-to-host communication Packets in IP layer are called datagrams. *Figure 5* shows the IP datagram format:

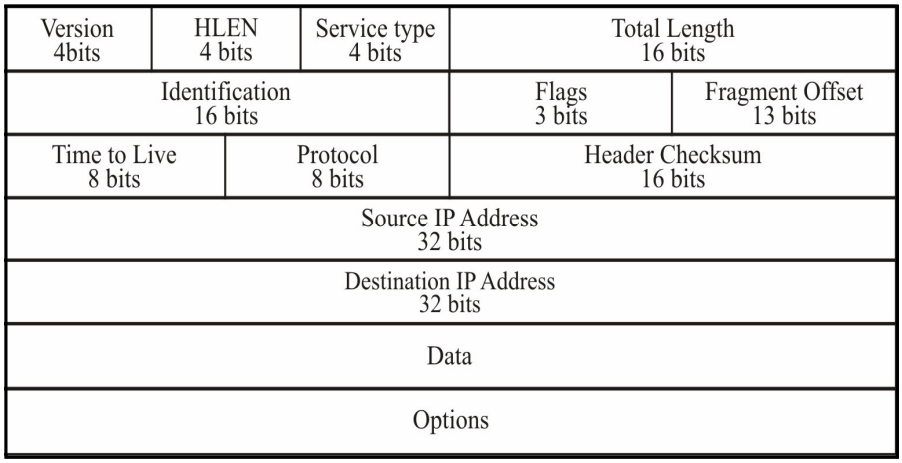


Figure 5: IP datagram

A brief description of header fields in order is given below:

- **Version (4 bits):** It defines the version of IP protocol. Currently, the version is 4(IPv4), indicated by value 4. In future it would contain 6, for IPv6.
- **HLEN (4 bits):** It is needed because length of header is variable. When the header size is 20 bytes, its value is 5(5*4=20). With options, the maximum size is 60 bytes, when the value is 15 (15*4=60). Each value represents number of 32-bit words.
- **Service Type (8 bits):** It is used to define type of service in terms of reliability, precedence, delay and cost.
- **Total length (16 bits):** it defines the total length of IP datagram. The maximum value can be 2^{16}=65536 bytes.
- **Identification (16 bits):** This field is used to unusually identify a datagram. It is useful to know the fragments belonging to same datagram fragments that are part of a datagram which contain same value in identification fields, so that they can be put together in the order to **reassemble** the datagram at receiver.
- **Flags (3 bits):** This field is used to uniquely identify a datagram. It is useful to know the fragments belonging to same datagrams.
- **Fragmentation Offset (13 bits):** It is a pointer that indicates the offset of the fragment in the original datagram before fragmentation.
- **Time to Live (8 bits):** It is used to control the maximum number of hops visited by the datagram. It is needed to restrict a datagram from continuing to travel in infinite loop without reaching the destination. This infinite looping may cause network congestion. This field limits the lifetime of datagram, after which the packet is discarded, so that datagram does not travel in infinite loop.

- **Protocol (8 bits):** An IP datagram may encapsulate data from various higher-level protocols like TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IP datagram should be delivered. Each protocol TCP, UDP etc. identified with a unique number.
- **Source Address (32 bits):** It stores the IP address of the source.
- **Destination Address (32 bits):** It stores the IP address of the final destination.
- **Options:** This field contains optional information such as routing details, timestamp etc. For instance, it can store route of a datagram, in the form of IP addresses of intermediate routers, optionally the time when it pass through that router.

The functions performed at this layer are as follows:

- **Define the datagram, which is the basic unit of transmission in the Internet:** The TCP/IP protocols were built to transmit data over the ARPANET, which was a *packet switching network*. A *packet* is a block of data that carries with it the information necessary to deliver it in a manner similar to a postal letter that has an address written on its envelope. A packet switching network uses the addressing information in the packets to switch packets from one physical network to another, moving them towards their final destination. Each packet travels the network independently of any other packet. The *datagram* is the packet format defined by IP.
- **Define the Internet addressing scheme:** IP delivers the datagram by checking the destination address in the header. If the destination address is the address of a host on the local network, the packet is delivered directly to the destination. If the destination address is not on the local network, the packet is passed to a router for delivery. Router is a devices that switch packets between the different physical networks. Deciding which router to use is called *routing*. IP makes the routing decision for each individual packet.
- **Move data between the Network Access Layer and the Host-to-Host Transport Layer:** When IP receives a datagram that is addressed to the local host, it must pass the data portion of the datagram to the correct host-to-host transport layer protocol. This selection is done by using the *protocol **filed*** in the datagram header. Each host-to-host transport layer protocol has a unique number that identifies it to IP.
- **Route datagrams to remote hosts:** Internet gateways are commonly (and perhaps more accurately) referred to as IP routers because they use IP to route packets between networks. In traditional TCP/IP jargon, there are only two types of network devices: gateways and hosts. Gateways forward packets between networks and hosts do not. However, if a host is connected to more than one network (called a *multi-homed host*), it can forward packets between the networks. When a multi-homed host forwards packets, it acts like any other gateway and is considered to be a gateway.
- **Fragment and reassemble datagrams:** As a datagram is routed through different networks, it may be necessary for the IP module in a gateway to divide the datagram into smaller pieces. A datagram received from one network may be too large to be transmitted in a single packet on a different network. This condition only occurs when a gateway interconnects dissimilar physical networks. For example, a physical network may be using Ethernet which specify packet (Frame) size of main 1500 bytes. Other physical network may be **using** FDDI that specify packet size **mark** of 4464 bytes. Therefore, the MTV of Ethernet to 1500 & MTU of FDDI is 4464. An IP packet-being transmitters over Ethernet is fragments limit 1500 bytes each.

Each type of network has a *maximum transmission unit (MTU)*, which is the largest packet it can transfer. If the datagram received from one network is longer than the

other network's MTU, it is necessary to divide the datagram into smaller fragments for transmission. This division process is called *fragmentation.*

1.5.6 Address Resolution Protocol (ARP)

We have seen that IP address makes the addressing uniform on the Internet. Routing of packets is done using the IP addresses of the packet. However, communication in a local network is broadcast, which is done using physical address. Therefore, when the packet reaches the destined network, there must be a process of obtaining the physical address corresponding to its IP address, of a computer in order to finally deliver the datagram to the destined computer. The physical address corresponding to an IP address is resolved by using address resolution protocol (ARP). ARP maps given IP address to a physical address as shown in the *Figure 6.* It takes host's IP address as input and gives its physical address as output.

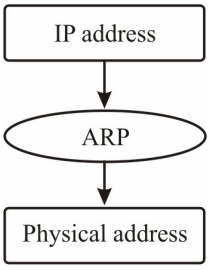


Figure 6: ARP maps the IP address to the physical address

ARP assumes that every host knows its IP address and physical address. Any time a host needs to know the physical address of another host on the network, it creates an ARP packet that includes the IP address X of the destination host asking–Are you the one whose IP address is X? If yes, please send back your physical address. This packet is then broadcasted over the local network. The computer, whose IP address matches X, sends a ARP reply packet, with its physical address. All the other hosts ignore the broadcast. Next time the host needs to send a datagram to the same destination, it need not broadcast an ARP query datagram; instead it can look up in its ARP cache. If the mapping is not found in the cache, then only the broadcast message is sent.

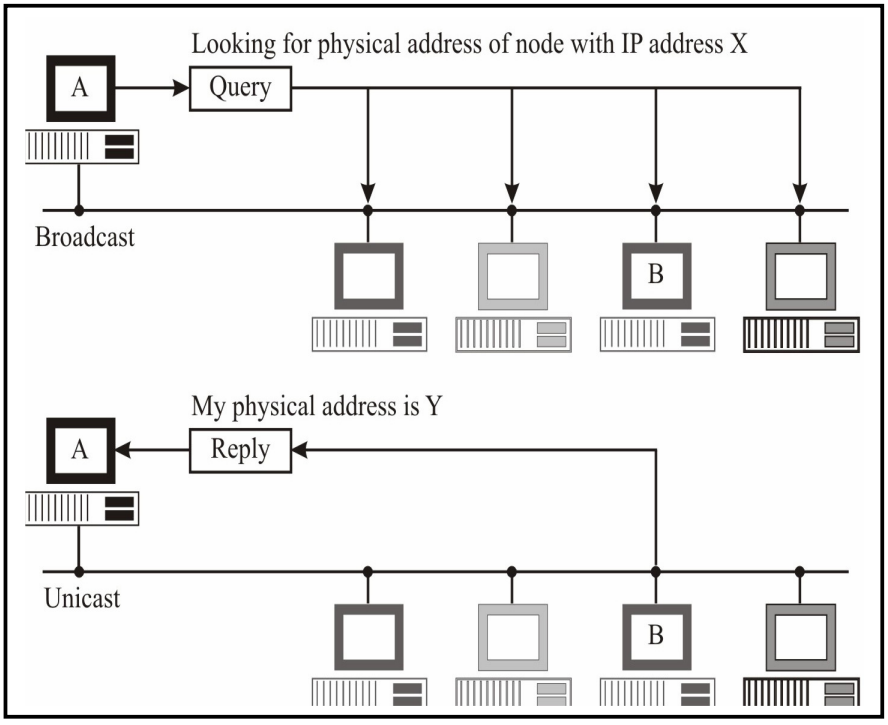


Figure 7: ARP query and reply

1.5.7 Reverse Address Resolution Protocol (RARP)

This protocol performs the job exactly opposite to ARP.It maps a physical address to its IP address as shown in *Figure 8*. Where is this needed? A node is supposed to have its IP address stored on its harddisk. However, there are situations when the host may not have hard disk at all, for example a diskless workstation. But also when a host is being connected to the network for the first time, at all such times, a host does not know its IP address. In that case RARP find out the IP address, this process is shown in *Figure 9*.

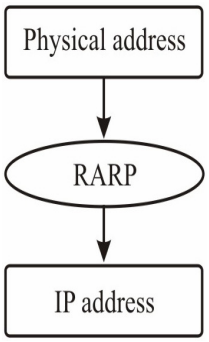


Figure 8: RARP maps the physical address to the IP address

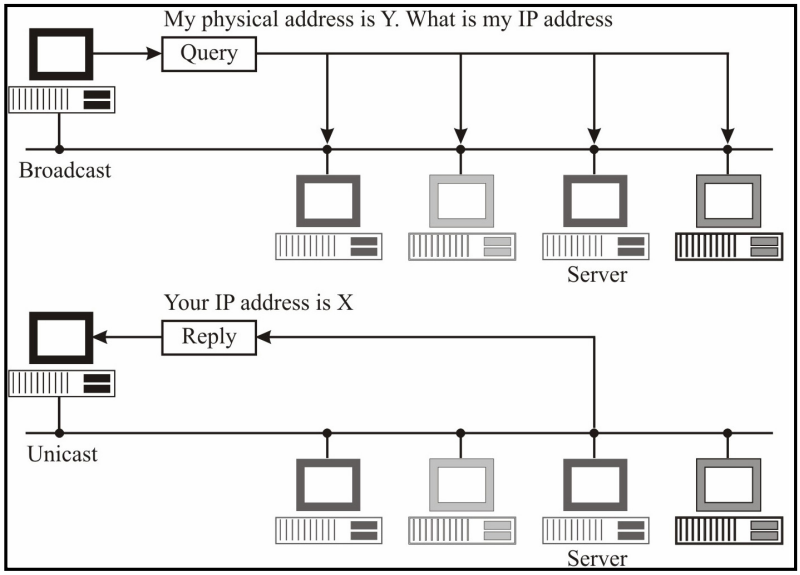


Figure 9: RARP query & reply

1.5.8 Internet Control Message Protocol (ICMP)

IP is best effort data delivery protocol. It means that IP makes best effort of delivering a datagram from source to destination. However, it does not guarantee that the datagram would be delivered correctly. IP has no error-reporting mechanism. A router or the final destination needs to inform the source if it must discard a datagram. IP also lacks a mechanism for host and management queries. A host sometimes needs to determine if a router or another host is alive to avoid sending datagrams to a router which is down. ICMP has been designed to report error occurred doing delivery of datagram.

The errors reported by ICMP are generally related to datagram processing. ICMP only reports errors involving fragment 0 of any fragmented datagrams. The IP, UDP or TCP protocols will usually take action based on ICMP messages. ICMP generally belongs to the IP layer of TCP/IP but relies on IP for support at the network layer. ICMP messages are encapsulated inside IP datagrams.

ICMP will report the following network information:

- Timeouts
- Network congestion
- Network errors such as an unreachable host or network.

The ping command is also supported by ICMP, and this can be used to debug network problems.

ICMP Messages

The ICMP message consists of an 8-bit type, an 8-bit code, an 8-bit checksum, and contents which vary depending on code and type.

ICMP is used for many different functions, the most important of which is error reporting. Some of these are *"port unreachable", "host unreachable", "network unreachable", "destination network unknown", and "destination host unknown"*. Some not related to errors are:

- **Timestamp request and reply** allows one system to ask another one for the current time.
- **Address mask and reply** is used by a diskless workstation to get its subnet mask at boot time.
- **Echo request and echo reply** is used by the ping program to see if another host is reachable. Thus two types of ICMP messages are defined: Error messages and Query messages.
- **Source Quench Message:** When datagrams arrive too quickly for processing, the destination host or an intermediate gateway sends an ICMP *source quench message* back to the sender. This message instructs the source to stop sending datagrams temporarily.
- **Redirect routes:** A gateway sends the ICMP *redirect message* to tell a host to use another gateway, presumably because the other gateway is a better choice. This message can only be used when the source host is on the same network as both gateways.

1.6 TRANSPORT LAYER

The transport layer runs on top of the Internet layer and is concerned with process-to-process delivery of data packets. Here, process is a running application program on a host. The main task of transport layer is to ensure correct delivery of packets. This introduces several responsibilities like flow control mechanism. By flow control we mean that a faster sender must not drawn a slow receiver with data packets resulting in data loss. The transport layer also provides connection mechanism. It establishes connection with the receiver, transfers data, and terminates the connection. The transport layer includes acknowledgement service to check for packet loss in the network. In TCP/IP transport layer is represented by two protocols: TCP and UDP.UDP is simpler of the two protocols. It is unreliable, connectionless transport protocol. Unreliability means that it does not provide any flow control, error control, and acknowledgement services. Connectionless means that no connection is established between the sender and the receiver prior to sending data. If UDP detects an error, it silently drops the packet. What is the use of UDP if it is so unreliable? It is very useful when the speed of the delivery is critical and loss of a small number of packets is tolerable like images, voice or video. TCP is useful when it is necessary to ensure error free delivery of packets from source to the destination.

1.6.1 Transmission Control Protocol

We have seen that IP packets may travel through different routes and may arrive out of sequence. TCP puts the packets in sequence. An intermediate router may discard the IP packets and they may not arrive at the destination at all. TCP checks for missing packets and handles this issue retransmission request. Some packets may get duplicated due to hardware malfunction. TCP discards duplicate packets. Hence, TCP takes care of all these situations and makes the Internet reliable. Before discussing TCP and UDP in detail, let us discuss process to process communication.

Process to Process Communication

IP provides host-to-host communication. TCP provides process-to-process communication using the client server paradigm as shown in *Figure 10*, suppose out of several application programs running on the host, TCP delivers the data packet to the destined application program.

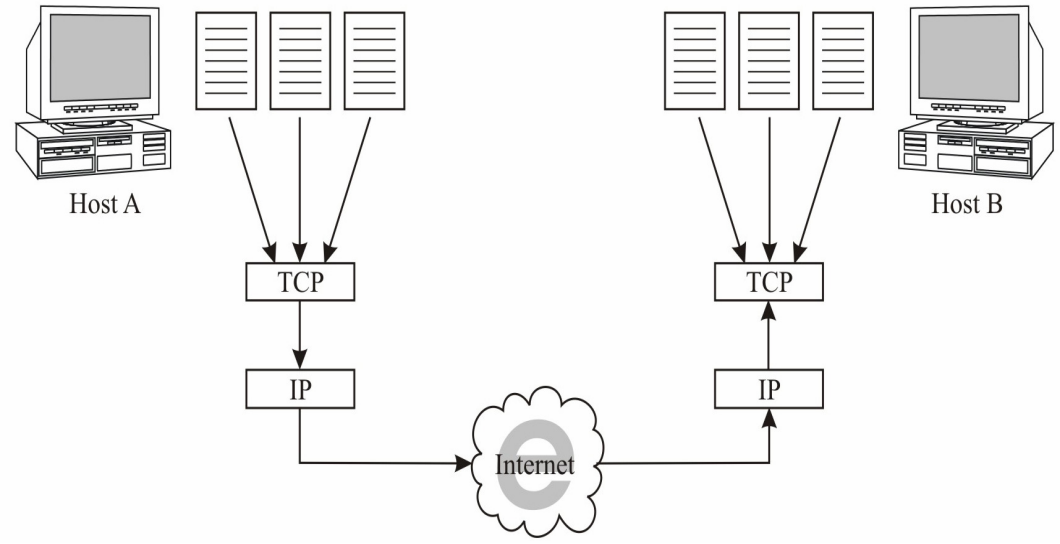


Figure 10: Domain of IP and TCP/UDP TCP like a multiplexer

Applications running on different hosts communicate with TCP with the help of concepts called ports. A port is a unique 16-bit number allocated to an application program. It is used when an application wants to open a TCP connection with another application on a remote computer. To understand how, let us take an analogy. A wants to call B staying in a hotel X, room no.Y. Now A dials the phone number of the hotel, after getting a response from the operator, A must tell the room number of B to be able to connect to B.This information is used to redirect the call to B.

Now, suppose an application X on host A wants to communicate to application Y on host B as given in *Figure 11* below. It must first reach B and then Y. Reaching B is not enough because there may be multiple applications running on B which might want to communicate with another application running on some other host. Hence, specifying the host is not enough.

A process on local host, called client, needs some services from a process on remote host called server. The client uses a random port number called ephemeral port number. The server also defines a port number with itself, which is known to all, and not random. This is because a client wanting to access a particular service will not know the port number if it is random. Therefore, TCP uses well-known port numbers. This also allows multiple TCP connections between two hosts are possible. Although the IP address would be same, the port number would be different each time.

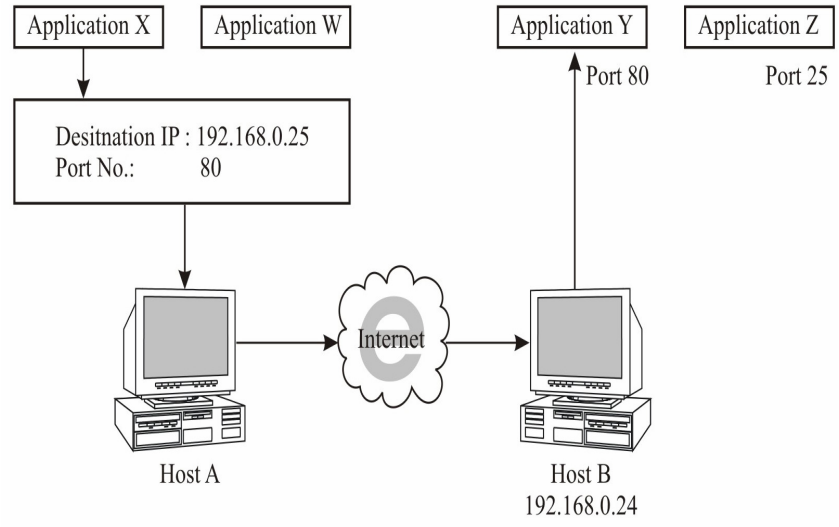


Figure 11: Use of port

The application X now provides the IP address of B, to identify the host B from thousands of hosts on the Internet. It provides port number. corresponding to application Y running on B to uniquely identify the application.

Sockets

The term Socket is used to identify the IP address and Port number concatenated together as shown in *Figure 12*.

Socket=IP address +port number

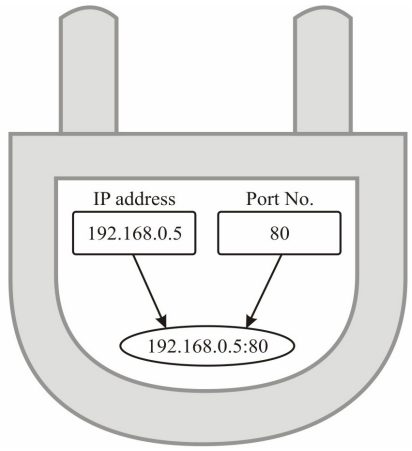


Figure 12: Socket

Socket Address

A client socket identifies a client process uniquely and a server socket identifies server process uniquely. Hence, a pair of sockets identifies a TCP connection between two applications on two different hosts.

Services Provided by TCP

- **Process to process communication**: TCP provides process-to-process communication as discussed above in section 1.6.1.
- **Stream Delivery Service**: TCP is stream oriented protocol. It allows the sending process to deliver data as a stream of bytes and allows receiving process to obtain data as a stream of bytes. For this, TCP needs two buffers, the sending buffer, and the receiving buffer, one for each direction. *Figure 13* shows the use buffers in TCP stream delivery.

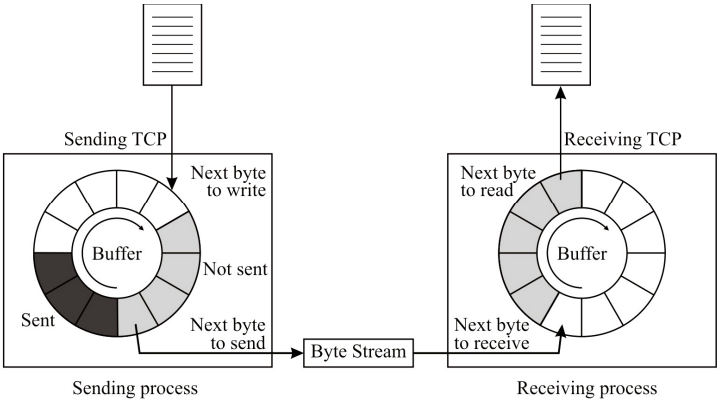


Figure 13: Stream Delivery Service

At the sending site, the buffer can have three types of chambers. The white section is the section of empty bytes which can be filled by the sender. The gray section contains bytes that have been sent but not acknowledged. They are kept in the buffer until the ACK is received. The black area contains the bytes to be sent by the sending TCP.

At the receiving site, the buffer is divided into two sections. The white section is the section of empty bytes which can be filled by the received bytes. The black section contains the bytes which can be read by the receiving process. It handles the speed mismatch between the sender and the receiver. There is one more step before the data can be sent.

IP, the service provider of TCP, sends data as packet, and not as stream of bytes. TCP groups a number of bytes into a segment, add a TCP header and delivers the packet to the IP layer for transmission. These segments can be of different sizes. The segment is now encapsulated inside a IP datagram and transmitted.

- **Connection Oriented Service:** When a process host A wants to communicate to a process on host B, the two TCPs establish a connection, data is exchanged, and the connection is terminated. This connection is virtual and not physical. All the IP datagrams may follow different routes, may arrive out of sequence. TCP take the responsibility of delivering the bytes in order. Hence maintaining a virtual connection. TCP provides full duplex communication. Data can flow in both the directions.

- **Reliability: TCP** is a reliable transport protocol. TCP incorporates error control, flow control, and acknowledgement services.

TCP Connections

We have said again and again that TCP is connection oriented. You may wonder how TCP connections orients as it uses the services of IP, a connectionless service. It is because TCP connection is virtual and not physical. It uses the services of IP for packet delivery but controls the connection itself. It takes care of reordering, retransmission, duplication, of which IP is unaware.

Let's see this connection establishment in detail.

TCP uses a technique called **three-way handshaking**. It means that three messages are exchanged between the sender and the receiver to establish the connection as shown in *Figure 14*. After this, TCP guarantees a reliable data transfer. It has been roved that this three way handshake is necessary and sufficient for establishing a successful connection.

The server waits passively to accept any active connection request. The client always initiates the connection request. The client sends the SYN segment in which only the SYN flag is set. This segment is used for the synchronisation of sequence numbers. The client chooses a random number and sends it as the first sequence number. It does not carry any relevant data, and consumes one sequence number.

The server sends the SYN ACK segment, with SYN and ACK bits set by sending this, server informs the client about the sequence number it will use in future and acknowledges the receipt of first SYN segment. It also consumes one sequence number.

The client sends the third segment, an ACK segment. It is an acknowledgement for the second segment. It bears the same sequence number as the first segment and hence, does not consume any sequence number. After this handshake, communication can start between the two ends.

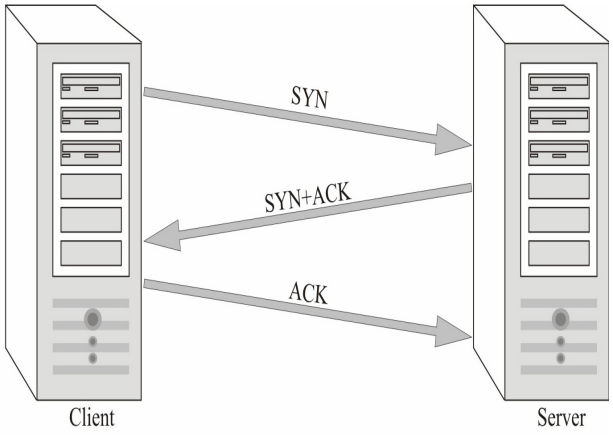


Figure 14: Three-way handshaking

The format of a TCP segment is shown in *Figure 15*. The header size is between 20 to 60 bytes, followed by data. It contains 20 bytes if the header does not contain any option.

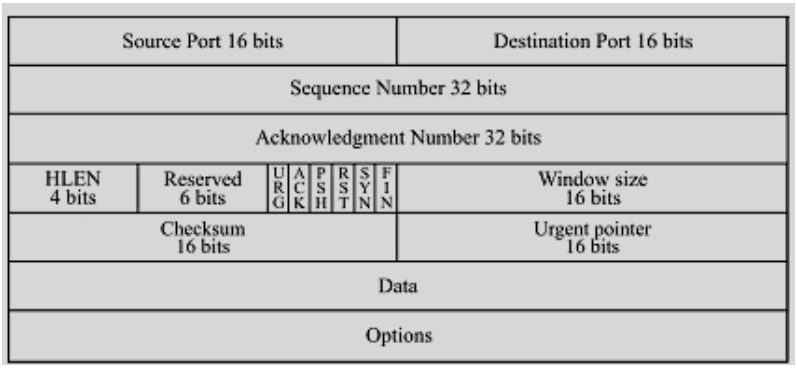


Figure 15:TCP segment format

Here is a brief description of header fields in order.

Source port number (16 bit): It specifies the source port number corresponding to the application which is sending the segment.

Destination Port Number (16 bit): It specifies the port number of the destination computer, corresponding to the receiving applications.

Sequence Number (4 bytes): It specifies the number assigned to the first byte of the data portion contained in this TCP packet. Each byte to be transmitted is numbered in

an increasing sequence. It tells the destination host which byte comprises the first byte of the TCP segment. During the connection establishment phase, the source and the destination generate a unique random number. If this random number is 3000, and the first segment contains 2000 bytes, then the sequence no. will be 3002. 3000 and 3001 are used in connection establishment. The second segment would have a sequence number of 5002(3002+2000), and so on.

Acknowledgement Number (4 bytes): On receiving a packet with sequence number X, the receiver sends back X+1 as the acknowledgement number. It defines the sequence number which the receiver is expecting next.

Header length (4 bits): The header length can be between 20 to 60 bytes. therefore, the value of this field can be between 5 (5*4=20) and 15(15*4=20)

Reserved: These 6 bits are reserved for future use.

Flag (6 bits): This field signifies 6 control flags, each one of them occupying one bit. Out of these, two are most important. The SYN flag indicate that the source wants to establish a connection with the destination. The FIN flag means that the sender wants to close the TCP connection.

Window Size (2 bits): This field determines the size of the window the other party must maintain. It is useful for flow control.

Checksum (16 bits): It contains the checksum for error detection.

Urgent Pointer: This field is used in situations when the segment contains urgent data. It specifies the number that must be added to obtain the number of the first urgent byte in the data section of the segment.

Pseudoheader: It is the part of the header of the IP packet in which the segment is encapsulated with some fields set to zero. It is added to have better error control. This way we ensure that if the IP header is corrupted, the segment is not delivered to the wrong host.

The protocol field is added to ensure that the packet belongs to TCP and not UDP because a process can use either TCP or UDP, the destination port number can be same. The value of this field is 6.If this value is changed, the checksum calculation will detect it and discard the packet instead of delivering it to the wrong protocol.

1.6.2 User Datagram Protocol (UDP)

The other protocol in transport layer is UDP. UDP is connectionless protocol. It allows a computer to send data without needing to establish a virtual connection. There is no error checking except for checksum. It does not provide sequencing, flow control or acknowledgements. Thus, UDP packets may be lost, arrive out of sequence, or duplicated. It is left to the application program to take care of these issues.

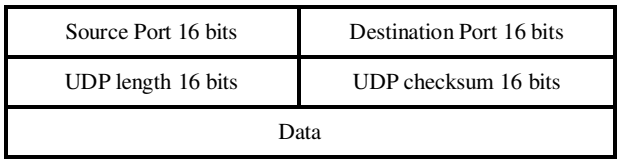


Figure 16: UDP packet format

Here is a brief description of header fields of DP packet is shown in *Figure 16*.

Source port number (16 bit): It specifies the source port number corresponding to the application which is sending the segment.

Destination Port Number (16 bit): It specifies the port number of the destination computer, corresponding to the receiving applications.

Total Packet Length (16 bits): It defines the total length of the UDP packet. However, this field is redundant, because there is a packet length field in IP, which encapsulates the UDP packet. Therefore UDP packet length=IP packet length-IP header length. This field is retained as an additional check.

Checksum: It is again used for error detection.

Pseudoheader: This field is added for the same purpose as in TCP.

☞ Check Your Progress 2

1) What are the functionalities of Internet layer?

……

……

……

……

……

……

……

2) What do you mean by ICMP source quench message

……

……

……

……

……

……

……

1.7 APPLICATION LAYER

Till now, we have studied Internet layer and Network layer protocols which provide end-to-end delivery of packets. These protocols would have no meaning without the Application Layer Protocols like DNS, HTTP, TELNET, and FTP. We will discuss all the application layer protocols in brief.

1.7.1 Electronic Mail

Email is one of the most popular Internet services. At the beginning of the Internet era, emails were short and consisted of text only. Today it is much more complex allowing to send text, audio and video. It also allows to send a message to more than one recipient.

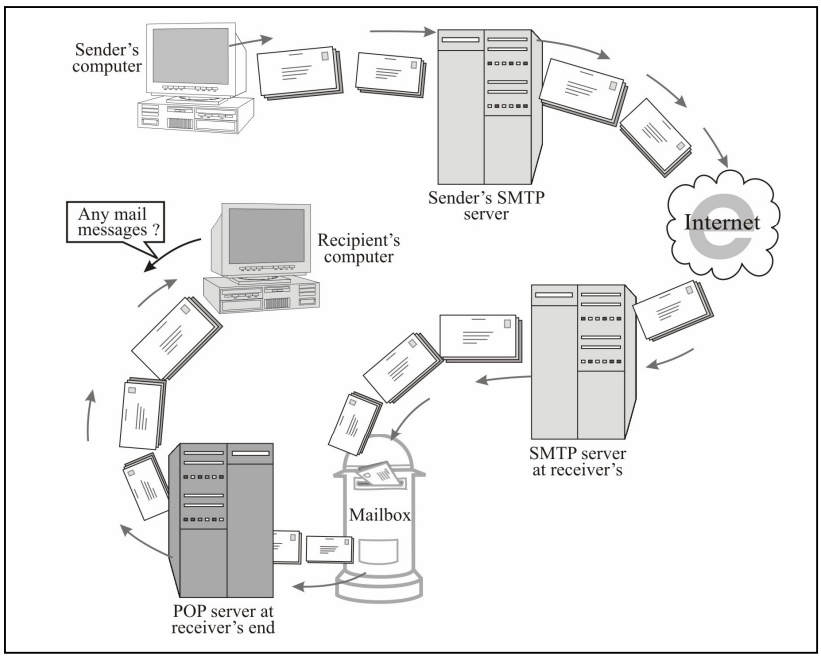


Figure 17: Journey of email message

In a typical scenario, both the sender and recipient are users on two different systems, and are connected to the system via a point to point WAN, or a LAN, which uses an email server for handling emails. Hence the message needs to be sent over the internet. How the sender computer sends the message through internet to the receipts computer is shown in *Figure 17*.

The sender uses an USER AGENT PROGRAM to prepare a message. It then sends the message through the LAN or WAN. This is done through a pair of message transfer agents (client and server). Whenever the sender has a message to send, he calls the user agent, which in turn, calls the Mail Transfer Agent (MTA) client. The MTA client establishes a connection with the MTA server, which is running all the time. The system at sender's site queues all the messages received. It then uses an MTA client to send message to the system at receiver's site. After the message arrives at Bob's mail server, another client server agents comes into picture which are called Message Access Agents (MAA). The receiver sends a request to the MAA server, which is running all the time, and requests the transfer of messages.

The email transfer protocols

For email messaging, every domain maintains an email server. The server runs protocols software that enable email communication. There are two main emails protocols: POP and SMTP. Because both the email protocol software programs run on server computers, the server computers are themselves called POP server and SMTP server. A single server can host both the POP and SMTP server programs.

SMTP server (Simple Mail Transfer Protocol)

SMTP is the Internet protocol used to transfer electronic mail between computers. The second generation of SMTP is called ESMTP (for Extended SMTP), but the differences are not important for this introduction.

It actually transfers the email message from the SMTP server of the sender to the SMTP server of the recipient. Its main job is to carry the message between the sender and the receiver. It uses TCP/IP underneath. That is, it runs on top of TCP/IP.

At the sender's site, an SMTP server takes the message sent by a user's computer.

The SMTP server at the sender's end then transfers the message to the SMTP server of the recipient.

The SMTP server at the recipient's end takes the message and stores it in the appropriate user's mailbox.

POP server

The Post Office Protocol provides a standard mechanism for retrieving emails from remote server for a mail recipient. Suppose that a home user X usually connects to the Internet using a dial–up connection to an ISP.Also, another person Y has sent an email to X, when X is not connected to the Internet. Now, the email message gets stored in the mailbox of the user provided by the ISP.

When X connects to the Internet next time and wants to see the new mails that have arrived for him since the last time he had connected to the Internet, he opens his email program. That email client program invokes POP client, which contacts the POP server. The email client opens the mailbox for the user X and sends the emails arrived for him to the POP client.

POP server is needed because SMTP server expects the destination host to be online all the time so that it can make a TCP connection to it, and forward the mail. But desktop computers are usually powered off after business hours. Hence the SMTP server forwards the mails to the mailbox, and the POP server retrieves the mails from the respective email boxes of the user when requested by POP clients.

POP3 (version 3). This is simple and limited in its functionality. It has two modes: the delete mode and the keep mode. In delete mode, mail is deleted from the mailbox after retrieval. It is used when the user is working at his personal computer and can save and organise the information after reading or replying. The keep mode is used when the user would check his mail from his primary computer. The mail is read but kept in the system for later retrieval.

IMAP Protocol

Another mail access protocol is the Internet Mail Access protocol, version 4(IMAP4).

It is similar to POP3, but it has more features as given below. In this, a user can-

- check the email header before downloading.
- search the contents of the email for a specific keyword prior to downloading.
- partially load the email it is helpful if the bandwidth is limited.
- create, rename, and delete mailboxes on the mail server.
- create a hierarchy of mailboxes in a folder for email storage.

File Transfer Protocol

We have seen how email works in *Figure 17*. But there are situations when we want to receive or send a file from or to a remote computer. Emails are just short messages. In fact, the greatest volume of data exchange in the Internet today is due to file transfer. Special software and a set of rules called File Transfer Protocol (FTP) exists for this purpose. It is an application layer protocol that is aimed at providing a very simple interface for any user of the Internet to transfer files. Whether you know it or not, you most likely use FTP all the time.

Although transferring files form one system to another seems simple and straight forward task, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different way to represent text or data. Two systems may have different directory structures. All these problems are elegantly solved by FTP.

FTP basics

FTP differs from other application layer protocols in one respect. All the other application layer protocols use a single connection between client and server. However, FTP uses two TCP/IP connections. One connection is used for actual data transfer, and the other is used for control information as shown in *Figure 18*. This makes FTP more efficient.

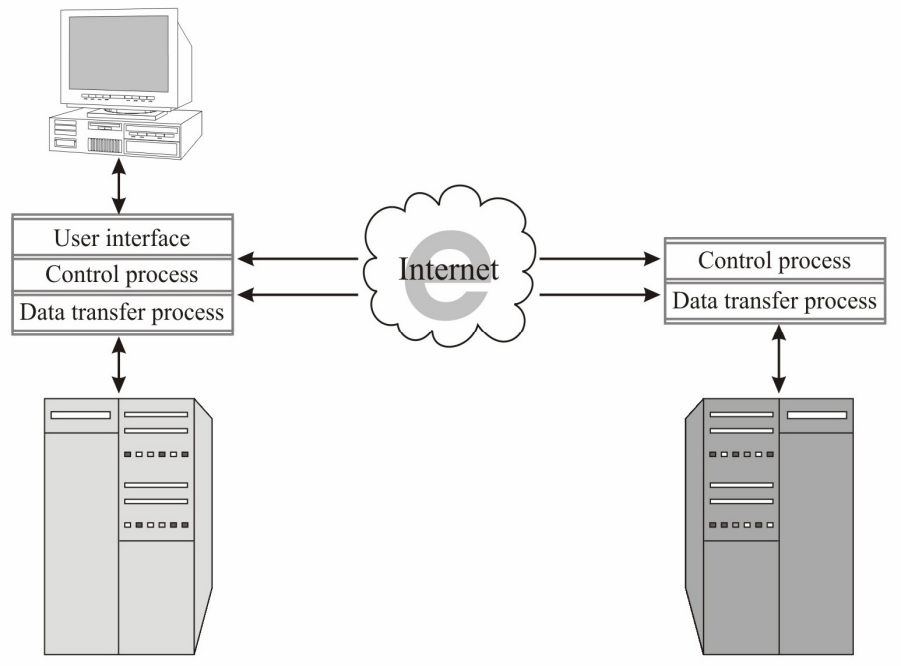


Figure 18: Data transfer through Internet

The components of the client and the server are shown in the *Figure 18*. The user interface component is not required at the server side, since there is no interaction exactly at the time of file transfer. The TCP control connection is made between the control processes of the client and the server. The TCP data transfer connection is made between the data transfer processes of the client and the server. While the file is being transferred, the server keeps track of how many bytes have been sent, and how much are remaining. It sends this information on the control connection, and this way reassures the user about the progress of data transfer. If multiple files are to be transferred in a single FTP session, then the control connection remains active throughout the entire FTP session. The data connection is opened and closed for each file transferred.

Prior to data transfer, the following attributes need to be specified:

Type of the file to be transferred: The file to be transferred can be an ASCIII, EBCDIC, or an image file. Image file is actually a misnomer. It has nothing to do with the image file. Instead, it signifies a binary file that is not interpreted by FTP in any manner, and it is sent as it is. If the file has to be sent as ASCII or EBCDIC, then the destination must be ready to accept in that mode. If the file has to be transferred without any regard to its content, then the third mode is used.

The structure of the data: File can be transferred by FTP by interpreting its structure in the flowing ways:

Byte-oriented structure: The file can be transferred as a continuous stream of data, where in no structure is assumed.

Record-oriented structure: The file is divided into records and the records are transferred one by one.

The transmission mode: FTP can transfer a file using any of the three modes as described.

Stream mode: This is the default mode; the data is delivered from FTP to TCP as a continuous stream of bytes.

Block mode: The data can be delivered from FTP to TCP as a block. Each data block follows a three byte header. The first byte is the descriptor, whereas the remaining to bytes defines the size of the block.

Compressed mode: The file can be compressed before sending. Normally Run Length Encoding is used for compressing a file.

This information is used by FTP to resolve the heterogeneity problem.

WU-FTP, CuteFTP is most commonly used FTP software applications.

What is an FTP Site?

An FTP site is like a large filing cabinet. With a traditional filing cabinet the person who does the filing has the option to label and organise the files, however they see fit. They also decide which files to keep locked and which remain public. It is the same with an FTP site.

The virtual 'key' to get into FTP site is the UserID and Password. If the creator of the FTP site is willing to give everyone access to the files, the UserID is 'anonymous' and the Password is your e-mail address (e.g. name@domain.com). If the FTP site is not public, there will be a unique UserID and Password for each person who is granted access.

When connecting to an FTP site that allows anonymous logins, you're frequently not prompted for a name and password. Hence, when downloading from the Internet, you most likely are using an anonymous FTP login and you don't even know it.

To make FTP connection you can use a standard Web browser (Internet Explorer, Netscape, etc.) or a dedicated FTP software program, referred to as an FTP 'Client'.

When using a Web browser for an FTP connection, FTP uploads are difficult, or sometimes impossible, and downloads are not protected (not recommended for uploading or downloading large files).

When connecting with an FTP Client, uploads and downloads couldn't be easier, and you have added security and additional features. For once, you're able to resume a download that did not successfully finish, which is a very nice feature for people using dial-up connections who frequently loose their Internet connection.

1.7.2 Domain Name System (DNS)

Machines on the Internet are identified by numerical "IP addresses" like 192.0.2.1. Domain names make it possible to refer to machines by a name rather than a number, which is definitely easier. If we had to remember the IP addresses of all of the Web sites we visit every day, we would all go nuts. Human beings just are not that good at remembering strings of numbers. In early days of Internet, all host names and associated IP addresses were recorded in a single file called hosts.txt, which was maintained by NIC (Network Information Center) in the US. Every night all the hosts attached to the Internet would obtain a copy of this file to refresh their domain name entries. As the internet grew, it became impossible to keep it up-to-date. To solve this problem, Domain Name System was developed. This DNS is consulted whenever any message is sent to any other computer on the Internet. It simply gives the mapping of domain names to IP addresses. Domain name servers translate domain names to IP addresses. That sounds like a simple task, and it would be able to handle four things:

- There are billions of IP addresses currently in use, and most machines have a human-readable name as well.
- There are many billions of DNS requests made every day. A single person can easily make a hundred or more DNS requests a day, and there are hundreds of millions of people and machines using the Internet daily.
- Domain names and IP addresses change daily.
- New domain names get created daily.

The DNS Name Space

The problem of mapping host names to IP addresses is very difficult given the numbers of machines on the Internet.

The postal system faces a similar challenge, which it has dealt by requiring the sender to specify the country, state, city, street name, house no. Internet uses the same principle. The Internet is divided into top-level domains, such having several hosts underneath. Again, each domain is divided into sub-domains, which can be further classified into sub-domains. This creates a tree-like structure which is shown in *Figure 19*.

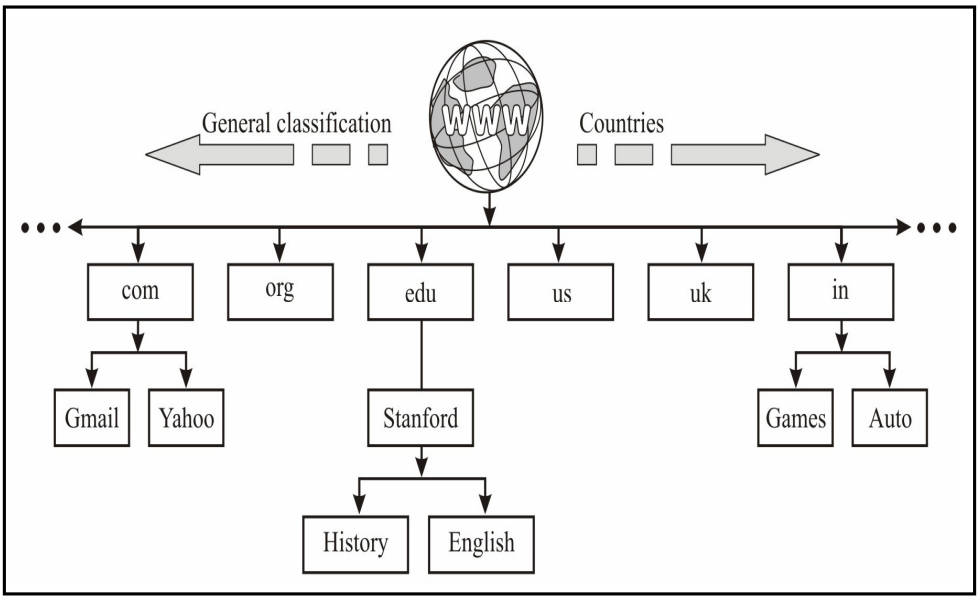


Figure 19: Domain Name space

The topmost domains are classified as: *generic,* and *countries*. The generic domains are classified into com (commercial), gov (the US federal government), edu (educational), org (non-profit organizations), net (network, etc. The country domains specify one entry for each country these clarifications are also shown in *Figure 20*. For example, uk (United Kingdom), in (India) and so on.

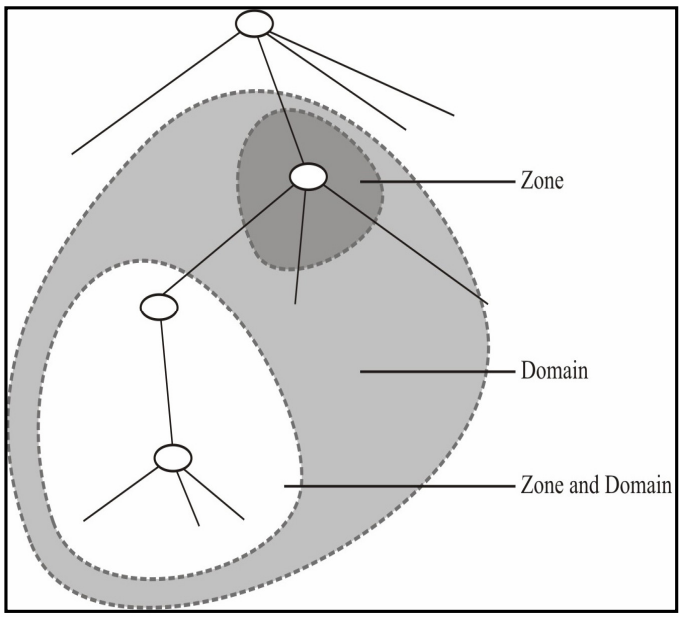


Figure 20: Zones and domains in DNS

For registering a name called *bajaj under the category auto, which is within* India(in), the full path will be bajaj.auto.in

Each domain is fully qualified by the path upward from it to the top most root. The names within a full path are separated by dot. Remember domain names are case-insensitive.

DNS Server

The information is contained in the domain name space. It must be stored at some place, but it would be inefficient and unreliable to store all the information at one place. The solution is to distribute the information among many computers called DNS servers. Thus, every domain has a domain name server.

Primary and Secondary DNS servers

DNS defines two types of servers: Primary server and secondary server.

A *primary server* is responsible for creating, maintaining and updating the zone file. It stores this zone file on its local disk.

A *secondary server* neither creates nor updates the zone file but stores the latest zone file from the primary server. The idea is to have redundancy so that if the primary server fails, the secondary server can continue serving the clients. A server can be primary server for one zone and secondary server for another zone.

1.7.3 How does the DNS Server Works?

DNS is designed to be a client server application. A host that needs to map a name to an address is called a revolver.

The DNS works in a manner similar a telephone directory inquiry service. You dial up the inquiry service and ask for a person's telephone number by providing his name. If the person is local, the inquiry service immediately comes up with the answer. If s/he stays in another state, the directory service directs your call to that state's telephone directory service, or asks you to call them. This is very similar to the way. The DNS server works. In this case, you specify the domain name and ask for its corresponding address.

All day, a DNS server does two things:

- Accepting requests from programs for mapping domain names into IP addresses (revolvers).
- Accepting requests from other DNS servers to map domain names into IP addresses.

When such a request comes, a DNS server has the following options:

- It can supply the IP address because it knows it from its zone file.
- It can contact another DNS server and try to locate the IP address for the name requested. Every DNS server has an entry called alternate DNS server, which is the DNS server it should get in touch with for unresolved domains. If this server is in authority, it responds, otherwise sends the query to another server. When the query is finally resolved, it travels back until it finally reaches the resolver. This type of resolution is called recursive resolution which is shown in *Figure 21* given below.

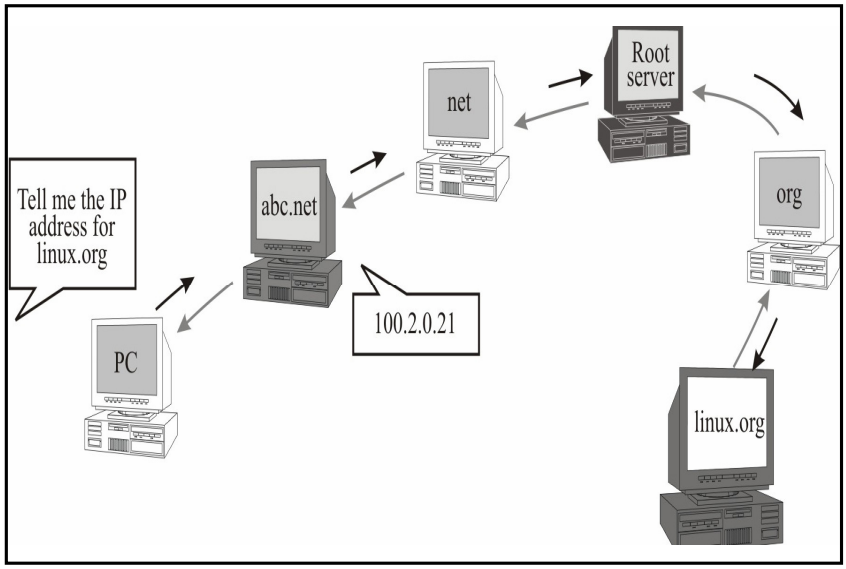


Figure 21: Recursive solution

It simply says I do not know the IP address for the requested domain but here is the IP address for a name server which knows more than me". The client is responsible for repeating the query to this second server. If this newly addressed server can resolve the query, it answers the query, and otherwise, it returns address of the alternate server. This process is called iterative resolution which is shown in *Figure 22*.

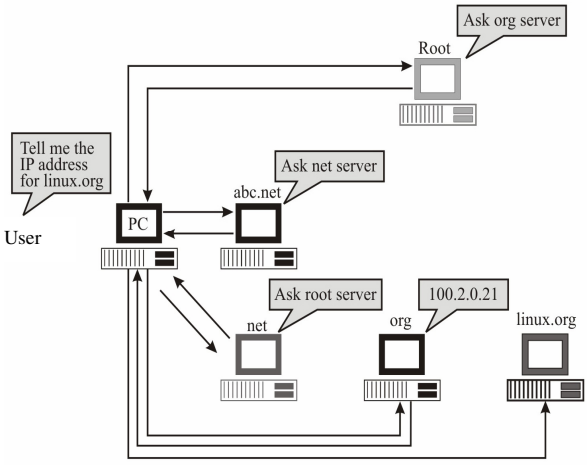


Figure 22: Iterative solution

It can return an error message because the requested domain name is invalid or does not exist.

Caching

Each time a server receives a query for the name that is not in its domain, it sends the query to the alternate server. It stores the information received from the response in its cache memory before sending it to the client. If any client asks for the same mapping, it an check its cache memory and resolve the problem. Thus, caching speeds up the resolution process to avoid sending outdated mapping to clients. The server adds an information called time-to live (TTL) to the cache entry. It define this time in seconds that the receiving server can cache the information. After that the entry is invalid and any query must be sent to the alternate server. DNS requires that each server keeps a TTL counter for each mapping it caches.

1.7.4 Simple Network Management Protocol (SNMP)

Since it was developed in 1988, the Simple Network Management Protocol has become the de facto standard for internetwork management. Because it is a simple solution, requiring little code to implement, vendors can easily build SNMP agents to their products. SNMP is extensible, allowing vendors to easily add network management functions to their existing products. SNMP also separates the management architecture from the architecture of the hardware devices, which broadens the base of multivendor support. Perhaps most important, unlike other so-called standards, SNMP is not a mere paper specification, but an implementation that is widely available today.

SNMP is based on the manager/agent model consisting of a manager, an agent, a database of management information, managed objects and the network protocol. The manager provides the interface between the human network manager and the management system. The agent provides the interface between the manager and the physical device(s) being managed (see the illustration above).

A typical agent usually:

- Implements full SNMP protocol.
- Stores and retrieves management data as defined by the Management Information Base.
- Can asynchronously signal an event to the manager
- Can be a proxy for some non-SNMP manageable network node.

A typical manager usually:

- Implemented as a Network Management Station (the NMS)
- Implements full SNMP Protocol Able to
- Query agents
- Get responses from agents
- Set’s variables in agents
- Acknowledge’s asynchronous events from agents.

The manager and agent use a **Management Information Base (MIB)** and a relatively small set of commands to exchange information. The MIB is organised in a tree structure with individual variables, such as point status or description, being represented as leaves on the branches. A long numeric tag or **object identifier** (OID) is used to distinguish each variable uniquely in the MIB and in SNMP messages.

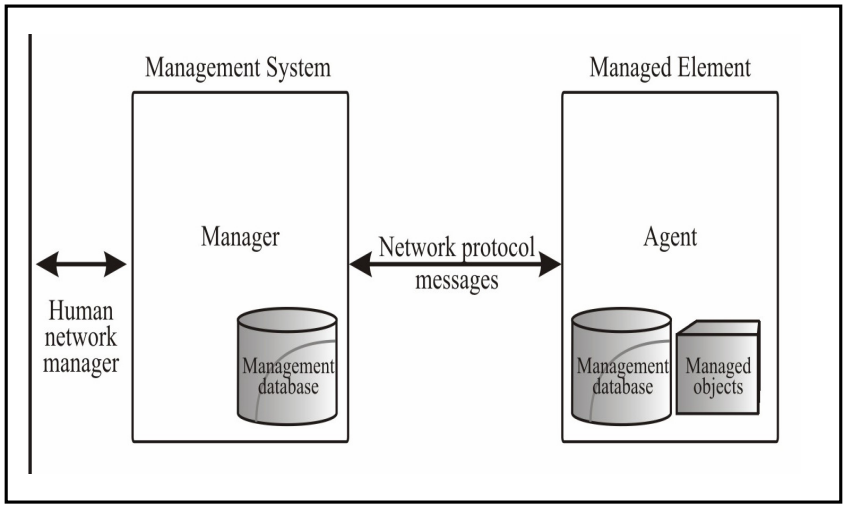


Figure 23: SNMP Protocol

SNMP uses five basic messages (GET, GET-NEXT, GET-RESPONSE, SET, and TRAP) to communicate between the manager and the agent. The GET and GET-NEXT messages allow the manager to request information for a specific variable. The agent, upon receiving a GET or GET-NEXT message, will issue a GET-RESPONSE message to the manager with either the information requested or an error indication as to why the request cannot be processed. A SET message allows the manager to request a change to be made to the value of a specific variable in the case of an alarm remote that will operate a relay. The agent will then respond with a GET-RESPONSE message indicating the change has been made or an error indication as to why the change cannot be made. The TRAP message allows the agent to spontaneously inform the manager of an 'important' event.

As you can see, most of the messages (GET, GET-NEXT, and SET) are only issued by the SNMP manager. Because the TRAP message is the only message capable of being initiated by an agent, it is the message used by DPS Remote Telemetry Units (RTUs) to report alarms. This notifies the SNMP manager as soon as an alarm condition occurs, instead of waiting for the SNMP manager to ask.

The small number of commands used is only one of the reasons SNMP is "simple." The other simplifying factor is its reliance on an unsupervised or connectionless communication link. This simplicity has led directly to its widespread use, specifically in the Internet Network Management Framework. Within this framework, it is considered 'robust' because of the independence of the managers from the agents, e.g., if an agent fails, the manager will continue to function, or *vice versa*. The unsupervised communication link does however create some interesting issues for network alarm monitoring.

Security levels with basic SNMP

Different Security levels (like authentication and authorization) are implemented in SNMP, let find out what we meant by authentication and authorization.

Authentication

Trivial authentication based on plain text community name is exchanged in SNMP messages. Authentication is based on the assumption that the message is not tampered with or interrogated.

Authorization

Once community name is validated then agent or manager checks to see if sending address is permitted or has the rights for the requested operation. "View" or "Cut" of

the objects together with permitted access rights is then derived for that pair (community name, sending address).

Underlying Communication Protocols

SNMP assumes that the communication path is a connectionless communication subnetwork. In other words, no prearranged communication path is established prior to the transmission of data. As a result, SNMP makes no guarantees about the reliable delivery of the data. Although in practice most messages get through, and those that don't can be retransmitted. The primary protocols that SNMP implements are the User Datagram Protocol (UDP) and the Internet Protocol (IP). SNMP also requires Data Link Layer protocols such as Ethernet or TokenRing to implement the communication channel from the management to the managed agent. The connectionless nature of SNMP leaves the recovery and error detection up to the NMS (Network Management Station) and even up to the agent. However, keep in mind that SNMP is actually transport independent (although original design was connectionless transport function, which corresponds to the UDP protocol) and can be implemented on other transports as well:

1.7.5 Remote Login: TELNET

The main task of the Internet and its TCP/IP protocols is to provide services for users. For example, a user may want to run an application at the remote site and create results that can be transferred to the local site. One way to satisfy this demand is to create client-server applications program for each such desired service. Though, file transfer programs (FTP), email (SMTP), etc. are available, but it would be impossible to write a client-server application for each demand. The better solution would be a general purpose client-server program that lets a user access any application program on a remote user, or allow a user to log on to a remote computer. TELNET (**TERminal NETwork**) allows remote login services.

TELENT has two parts: client and server. Once a user using the services of a TELNET client connects to the remote TELNET server the keystrokes type by the user on the client are sent to the remote server to be interpreted upon to give the impression that the user is working on the remote computer.

Local login

When a user logs on to a local time sharing system, it is called local login. As user types at a terminal, the keystrokes are accepted by the terminal driver and passed to the operating system. The operating system interprets the combination of characters, and invokes the desired application. This mechanism is not simple as the operating system may associate special meanings to special characters. For example, Ctrl+z means suspend in UNIX. This does not create problem in local login because the terminal emulator and the driver know the exact meaning of characters.

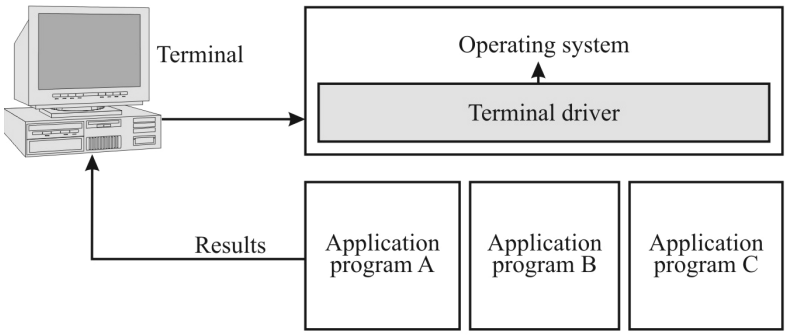


Figure 24: Local Login

Remote login

When a user logs on to a access an application on a remote computer, the users need to perform remote login.

Here the telnet client and server come into use. The user sends the keystrokes to the terminal driver where the local operating system accepts the characters but does not interpret them. The characters are sent to the TELNET client, which transform the characters to a universal character set called Network Virtual Terminal (NVT) and delivers them to the local TCP/IP stack. This is necessary because telnet is a general purpose application, and was designed to work between any terminal and any host. Thus, the client maps the terminal type to NVT. At the other end, server maps the NVT on to the actual terminal type the server is serving.

The commands or text, in NVT format travels through the Internet, and arrive at the TCP/IP stack of the remote machine. The characters are delivered to the local operating system, and are passed to the TELNET server, which performing mapping of NVT characters. However, they can be directly passed to the operating system, because it is designed to receive characters from terminal driver. The solution is to add a pseudo terminal driver, which pretends that the characters are coming from the terminal and not from the client. The operating system then passes the characters to the appropriate application.

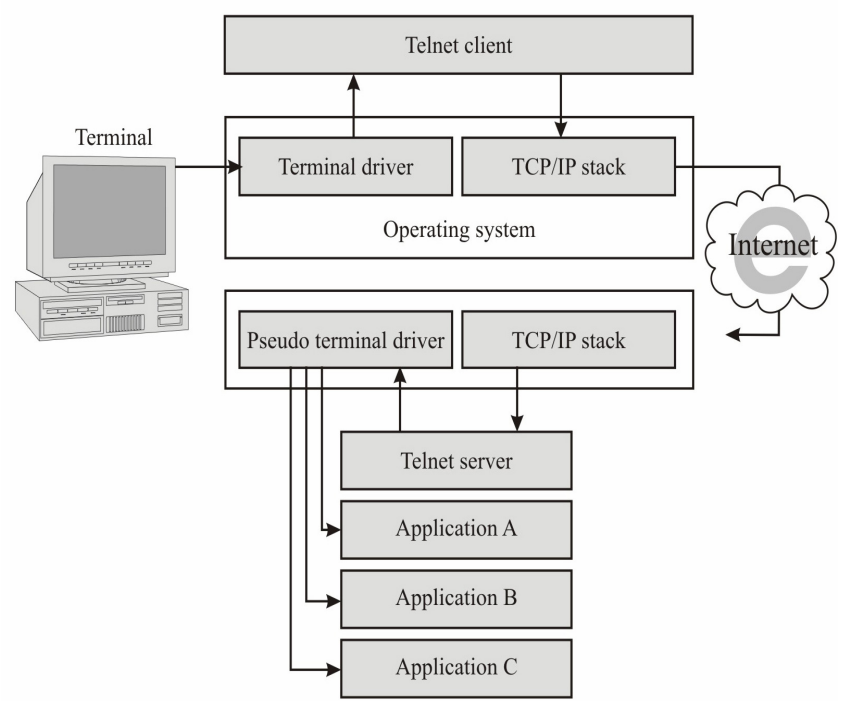


Figure 25: Remote Login

Communication in TELNET

Technically, TELNET server is quite complicated. It has to handle many clients at the same time, and respond in real time. To handle this issue, TELNET server uses the principle of delegation. Whenever there a new client request for a TELNET connection, it creates a child process and lets the child handle the particular client.

TCP uses only one TCP connection. The same TCP connection is used to transfer both the control and data characters. How does the TELNET then distinguish between control and data characters? For this, it mandates that each sequence of control characters must be preceded by a special control character called Interpret As Command (IAC).

1.7.6 World Wide Web: HTTP

The WWW project was initiated by CERN to create a system to handle distributed resources necessary for scientific research. Apart from email, the most popular applications running on Internet is the World Wide Web (WWW). It is so popular that people confuse it with Internet.

However, it is just an application such as email, FTP that uses TCP/IP. Many companies have internet websites, which is a collection of web pages stored on a web server. The server has web server software running on it. The function of web server is to store web pages and transmit them to a client computer when requested to do so.

In your company's website, you can display the information about the products, employees and policies. Sites can be dedicated to specialised tasks such as displaying news, share prices, sports etc. WWW consists of thousands of such websites of individuals and companies giving huge details about people, companies, events, news etc. WWW is an online repository of information that users can view using a program called web browser.

Architecture

The WWW today is a distributed client-server service. The client using a browser can access a service using a server. The service provider is distributed over many locations called sites.

Web server

A web server is a program running on a server computer, additionally, it consists of the web site, consisting of many web pages. It is simply a file written in HTML (Hypertext Markup Language). It can consist of text, graphics, sound, video, animation. Every web site has a server process that passively listens for TCP connection requests at port 80. After the connection is established, the client sends one request and the server sends one response. The request – response model is governed by Hyper Text Transfer Protocol (HTTP).

Web browser

A web browser acts as a client in WWW interaction. Using this program, user can request for a web page on a web server. The browser then interprets and displays the document. A variety of vendors offer commercial browsers. A browser usually consists of three parts: a controller, client protocol and interpreters. The controller receives the input from the keyboard or the mouse and uses client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on screen. The client protocol can be one of the protocols described previously like FTP, TELNET, or HTTP.

Uniform Resource Locator (URL)

A client that wants to access a web page needs to specify the address. The Uniform Resource locator is the standard for specifying any kind of information on the Internet. It has four things: protocol, host computer, port, and path.

Anatomy of URL

http://www.ignou.ac.in/80/index

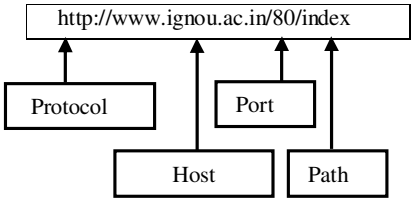


Figure 26: Components of URL

- *The protocol* is the client-server program used to retrieve the document. Most common is HTTP.
- *Host* is the name of the computer on which the information resides.
- Port number is transport address of the client or server program on a website.
- *Path* is the pathname of the file on the website, and can consists of slashes.

HTTP

Hyper Text Transfer Protocol (HTTP) is a used mainly to transfer data on World Wide Web. The commands from the client are embedded in a request message .The contents of the request message are embedded in a response message. HTTP uses the services of TCP at port 80.

HTTP is a stateless protocol since each transaction is independent of the previous transaction. The TCP connection between the client and the server is established for every page. It does not remember anything about the previous request. Keeping HTTP stateless was aimed at making the Web simple.

Sample HTTP request and response transaction is shown in figure 27.

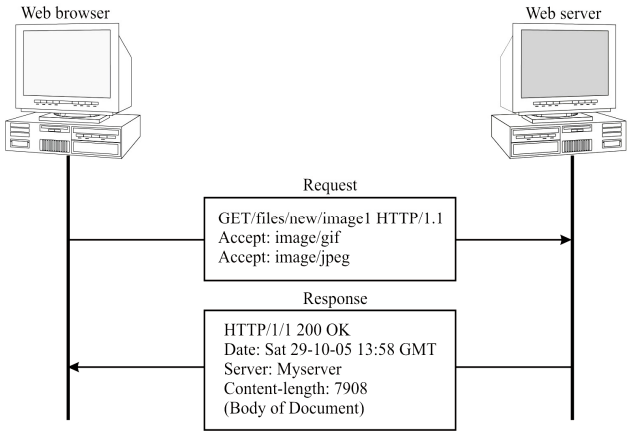


Figure 27: HTTP request and response

The GET command requests the web server at www.myserver.com for a image file image1. The HTTp/1.0 indicates that the browser uses the 1.0 version of the HTTP protocol.

The first line of response indicates that the server is also using HTTP version 1.0. The return code of 200 indicates that the server processed the request successfully. Request message can be of following types.

Method	Action
GET	Requests a document from the server
HEAD	Request information about the document but not the document itself; i.e. head of the HTML page
POST	Sends some information from client to server. It appends the data to the existing document.
PUT	Sends a document from to server. It replaces the existing document.
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Enquire about available options

Locating Information on the Internet

The amount of information available on the Internet is mind-boggling. However, finding the right information is usually very cumbersome. For enabling people to search information efficiently, search engines or used.

The search engines may continuously crawl through the Web Pages on the Internet and gather information about them to facilitate search for users in those Web pages.

Check Your Progress 3

1) Explain the concept of recursive and iterative resolution in of DNS?

……

……

……

……

……

2) How does FTP differ from other application layer protocols?

……

……

……

……

……

1.8 NETWORKING EXAMPLE: PUTTING IT ALTOGETHER

After seeing how communication takes place-using computers, we describe four scenarios to give a clear picture of it

Example 1: In the first example, there are just two computers, which are not part of any network as shown in *Figure 28*.

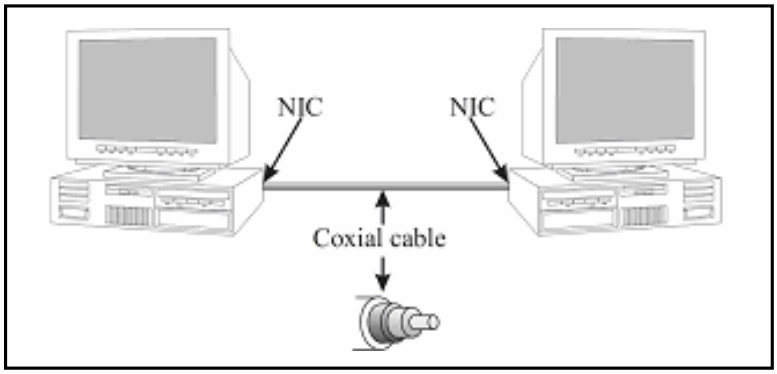


Figure 28: Two computers connected with Coaxer Cable

In order to have communication, both computers should have Network Interface card and TCP/IP software running over it. They can connect using coaxial cable, and can choose any IP address, as they are not part of any network. The link is very fast as there is no sharing of bandwidth.

Example 2:

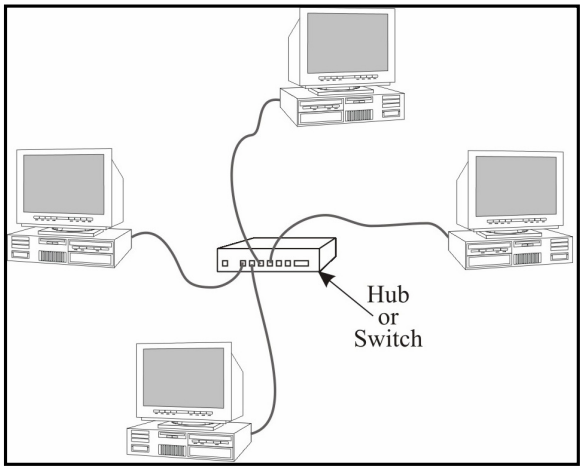


Figure 29: Computers connected with Hubs

A number of computers are connected to each other using a hub or a switch device as shown in *Figure 29*. Now the computers require uniform addressing, a network interface card and TCP/IP software they can connect to switch or hub using any physical medium. At a time only two computers can communicate. The packets from a computer are first transferred to the switch, which sends it to the destined computer.

Example 3:

A number of small networks like the ones in a computer Lab connect to each other, using routers as shown in *Figure 30*. A computer wanting to communicate to another computer first checks if the destination computer is on the same network. If so, it uses ARP to know the IP address of the destination and forwards the packet to it. However, if it is not on the same network, the packet is forwarded to the router connected to the network. A router is a multihome device, which is a part of more than one network. If the destination computer is connected to the router, it is delivered, otherwise it is forwarded to another router which is expected to know the way to destination. The path followed in order to route the packets is determined by the routers which employ criteria like shortest path, minimum delay, minimum cost to decide the next router to which the packet should be fowared. The links and thus the bandwidth is shared among the computers.

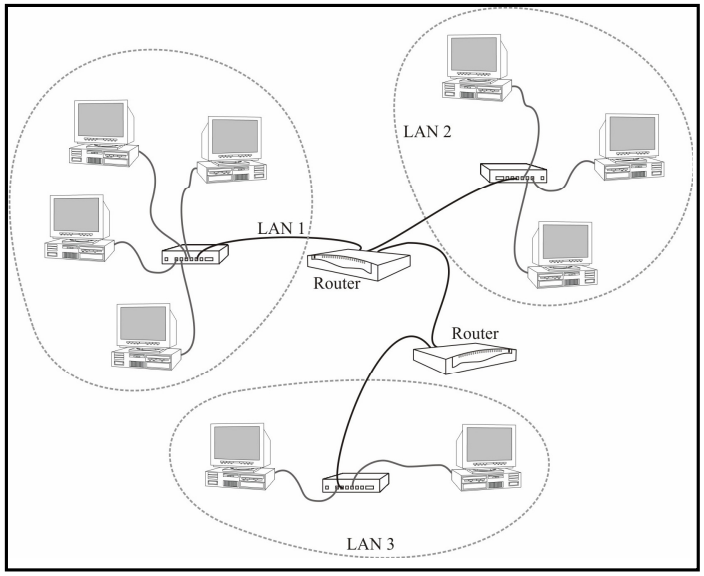


Figure 30: LANS connected with Routers

Example 4:

The fourth example is of the Internet given below in *Figure 31*, where a number of networks are connected to each other. These networks differ from each other in a number of ways like packet format, underlying hardware, and the protocols used. The scenario differs from the third one in that a gateway is required. A gateway can forward packets across different networks that may also use different protocols. Thus a gateway not only has the ability of translating between different packets formats, but also different protocols. Communication in the Internet is made possible by use of routers which forward packets between networks using the same protocol, and gateways connect large, incompatible networks.

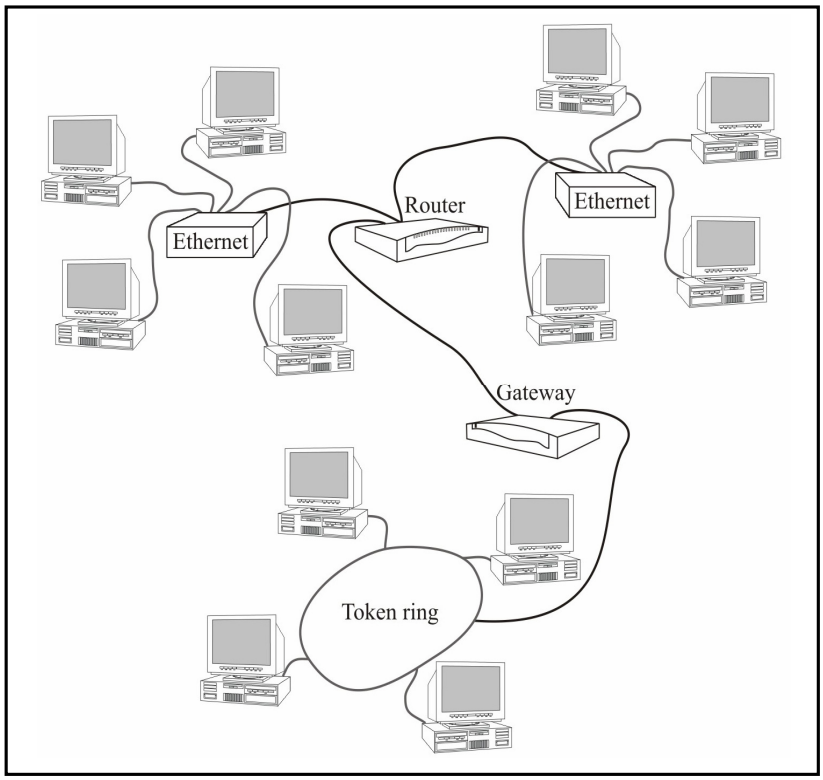


Figure 31: Different network connected with gateway and routers

1.9 SUMMARY

This unit is complete overview of TCP/IP protocols. Till now we have studied that the Internet and TCP/IP came into being at around the same time. Some standards need to be flowed at the hardware and software level in order to connect incompatible networks. TCP is the software standard for the Internet. It is organised as a stack of layers. Each layer performs has interface with adjacent layers and certain functionalities. IP is connectionless, unreliable, and best effort packet delivery mechanism. It provides node-to node communication ARP, RARP, ICMP are other Internet layer protocols. TCP is reliable, connection-oriented protocol, and uses the services of IP. UDP is another transport layer protocol, which is unreliable, connectionless. It is much simpler than TCP and used in applications, which do not require the reliability but needs faster delivery. Together they provide end-to-end delivery of packets. Application layer protocols provide users with mechanisms to use internet for file transfer, (FTP) connection to remote hosts (TELNET), viewing web pages (WWW), manage network resources (SNMP) and lot of other functionalities. In the end, scenarios for typical communication between two computers are discussed. The next unit of this course covers about the Internet Protocol and its role in TCP/IP.

1.10 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) The TCP/IP model is made up of four layers: interface layer, network, transport, and application. The first layer of TCP/IP (Application layer) is similar to the first three layers (Application, Presentation and Session layer) of the OSI model. The services of transport layers of both the models are similar. Further, the services of network layers in both models are also similar, while some time network layer is also known as Internet layer. The last layer of TCP/IP is Interface layer, which includes the services of data link layer and physical layer of OSI model. In OSI model, each layer takes the services of the lower layer. Whereas the layers of TCP/IP protocol suite contain relatively independent protocols.

2) The TCP/IP model is made up of four layers: interface, network, transport, and application. Layers of TCP/IP Protocol Suite are explained below in detail:

i) **Interface Layers**

The physical layer deals with the hardware level, voltages. The data link layer deals with media access and control strategies, frame format etc. At this level, TCP/IP does not define any protocol. It supports all standards and protocols.

ii) **Internet Layer or Network Layer**

The Internet layer is an important layer in the protocol suite. At this layer. TCP/IP supports Internetworking Protocol (IP). IP is a host-to host protocol. This layer is responsible for the format of data-grams as defined by IP, and routing a datagram or packet to the next hop, but is not responsible for the accurate and timely delivery of datagrams to the destination in proper sequence. IP allows raw transmission functions allowing user to add functionalities necessary for given application. An ensuring maximum efficiency, TCP/IP supports four other protocols: ARP, RARP, ICMP, IGMP.

iii) **Transport Layer**

At this layer, TCP/IP supports two protocols: TCP, UDP, IP is host-to-host protocol, which can deliver the packet from one physical device to another physical device. TCP, are UDP, are transport level protocols, responsible for delivering a packet from one process on a device to another process on the other device. User Datgram Protocol (UDP)

It is simpler of the two protocols. It does not provide reliability. It is, therefore faster, and using for applications in which delay is intolerable (in case of audio and video)

Transmission Control Protocol (TCP).

TCP is reliable, connection oriented protocol. By connection oriented, we mean that a connection must be established between both ends before either can transmit data. It ensures that communication is error-free and in sequence.

iv) **Application Layer**

As said earlier, it is closer to combined session, presentation, and application layer of OSI model. It allows user to run various applications on Internet. These applications are File Transfer Protocol (FTP), remote login (TELNET), email (SMTP), WWW (HTTP). The session layer of OSI model is almost dropped in TCP/IP.

Check Your Progress 2

1) Following are functions performed at the Internet layer:
   - Define the datagram, which is the basic unit of transmission in the Internet.
   - Define the Internet addressing scheme.
   - Move data between the Network Access Layer and the Host-to-Host Transport Layer.
   - Route datagrams to remote hosts.
   - Fragment and reassemble datagrams.

2) When datagrams arrive too quickly for processing, the destination host or an intermediate gateway sends an ICMP *source quench message* back to the sender. This message instructs the source to stop sending datagrams temporarily.

Check Your Progress 3

1) DNS server does two things one is accepting requests from programs for mapping domain names into IP addresses (revolvers) and another is accepting requests from other DNS servers to map domain names into IP addresses. When such a request comes, a DNS server has the following options.

i) It can supply the IP address because it knows it from its zone file.

ii) It can contact another DNS server and try to locate the IP address for the name requested. Every DNS server has an entry called alternate DNS server, which is the DNS server it should get in touch with for unresolved domains. If this server is in authority, it responds, otherwise sends the query to another server. When the query is finally resolved, it travels back until it finally reaches the resolver. This type of resolution is called ***recursive resolution.***

2) It simply says, I do not know the IP address for the requested domain but here is the IP address for a name server which knows more than me". The client is responsible for repeating the query to this second server. If this newly addressed server can resolve the query, it answers the query, and otherwise, it returns address of the alternate server. This process is called ***iterative resolution.***

3) It can return an error message because the requested domain name is invalid or does not exist.

1.11 FURTHER READINGS

1) Achyut S Godbole, *Web Technoligies*, TATA McGrawHill, 2003.

2) Berhouz Forouzan, *TCP/IP Protocol Suite*, 3rd edition, TATA McGraw Hill, 2006.

3) http://www.tcpipguide.com

4) http://www.cisco.com

UNIT 2 INTERNET PROTOCOL

Structure **Page Nos.**

2.0 Introduction 45

2.1 Objectives 45

2.2 Overview of Internet Protocol 46

2.3 IP Header 46

2.4 IP Address 50

2.4.1 IP Address Classes 50

2.4.2 Subnet Masks and CIDR Networks (Classless IP Addresses) 52

2.4.3 Internet-Legal Versus Private Addressing 55

2.5 IP Routing 57

2.5.1 Routing Protocol 57

2.5.2 Routing Algorithms 58

2.6 Summary 60

2.7 Answers to Check Your Progress 60

2.8 Further Readings 60

2.0 INTRODUCTION

Internet protocols were first developed in the mid-1970s, when the Defence Advanced Research Projects Agency (DARPA) became interested in establishing a packet-switched network that would facilitate communication between dissimilar computer systems at research institutions. The Internet Protocol (IP) is a network-layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is documented in RFC 791 and is the primary network-layer protocol in the TCP/IP protocol suite. Along with the Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols. IP has two primary responsibilities: providing connectionless, best-effort delivery of datagrams through an internetwork; and providing fragmentation and reassembly of datagrams to support data links with different maximum-transmission unit (MTU) sizes.

2.1 OBJECTIVES

Our objective is to introduce you with basic concept of Internet Protocol. On successful completion of this unit, you should be able to:

- have a reasonable understanding of the IP protocol architecture;
- describe the operation of IP protocol and its header format;
- understand the role and meaning of IP addressing and classes;
- describe and understand how to use subnet addressing; and
- understand the simple routing protocols.

2.2 OVERVIEW OF INTERNET PROTOCOL

Internet Protocol is the network (internet) layer protocol used by both TCP and UDP, transport layer protocols. The entire TCP or UDP datagrams (header + payload) travel through the network as a part of the IP datagrams. TCP or UDP datagram is encapsulated in IP datagram as illustrated in following *Figure 1*. It is also clear from the figure that the application layer packet is encapsulated in transport layer datagram and network layer datagram is encapsulated in link layer frame.

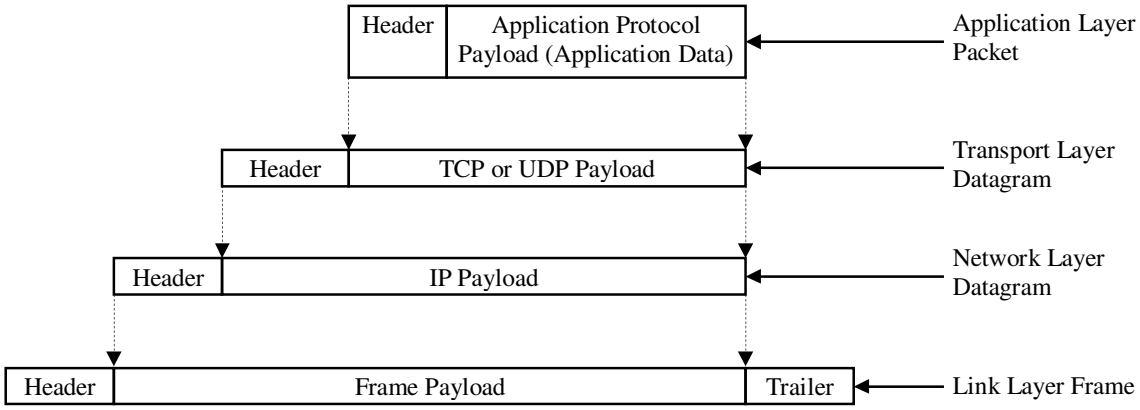


Figure 1: TCP/IP Protocols and Encapsulation

2.3 IP HEADER

The IP datagram consists of a header part and a data part. The IP header is six 32-bit words in length (24 bytes total) when all the optional fields are included in the header. The shortest header allowed by IP uses five words (20 bytes total). To understand all the fields in the header, it is useful to remember that IP has no hardware dependence but must account for all versions of IP software it can encounter (providing full backward-compatibility with previous versions of IP). The IP header layout is shown schematically in *Figure 2*. The different fields in the IP header are examined in more detail in the following subsections.

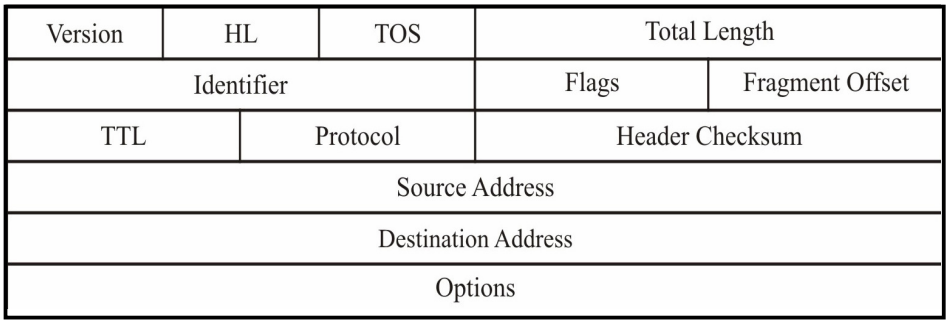


Figure 2: IPv4 Header Format

Version Number

This is a 4-bit field that contains the IP version number the protocol software is using. The version number is required so that receiving IP software knows how to decode the rest of the header, which changes with each new release of the IP standards. The most widely used version is 4, although several systems are now testing version 6 (called IPng). The Internet and most LANs do not support IP version 6 at present.

Part of the protocol definition stipulates that the receiving software must first check the version number of incoming datagrams before proceeding to analyse the rest of the header and encapsulated data. If the software cannot handle the version used to build the datagram, the receiving machine's IP layer rejects the datagram and ignores the contents completely.

Header Length

This 4-bit field reflects the total length of the IP header built by the sending machine; it is specified in 32-bit words. The shortest header is five words (20 bytes), but the use of optional fields can increase the header size to its maximum of six words (24 bytes). To properly decode the header, IP must know when the header ends and the data begins, which is why this field is included. (There is no start-of-data marker to show where the data in the datagram begins. Instead, the header length is used to compute an offset from the start of the IP header to give the start of the data block).

Type of Service

The 8-bit (1 octet) Service Type field instructs IP how to process the datagram properly. The field's 8 bits are read and assigned as shown in *Figure 3*, which shows the layout of the Service Type field inside the larger IP header shown in *Figure 2*. The first 3 bits indicate the datagram's precedence, with a value from 0 (normal) through 7 (network control). The higher the number, the more important the datagram and, in theory at least, the faster the datagram should be routed to its destination. In practice, though, most implementations of TCP/IP and practically all hardware that use TCP/IP ignore this field, treating all datagrams with the same priority.

Precedence (3 bits)	Delay	Thru	Rel	Not Used

Figure 3: Type of Service

The next three bits are 1-bit flags that control the delay, throughout, and reliability of the datagram. If the bit is set to 0, the setting is normal. A bit set to 1 implies low delay, high throughout, and high reliability for the respective flags. The last two bits of the field are not used. Most of these bits are ignored by current IP implementations, and all datagrams are treated with the same delay, throughput, and reliability settings.

For most purposes, the values of all the bits in the Service Type field are set to 0 because differences in precedence, delay, throughput, and reliability between machines are virtually nonexistent unless a special network has been established. Although these flags would be useful in establishing the best routing method for a datagram, no currently available UNIX-based IP system bothers to evaluate the bits in these fields. (Although it is conceivable that the code could be modified for high security or high reliability networks.)

Total Length (or Packet Datagram Length)

This field gives the total length of the datagram, including the header, in octets. The length of the data area itself can be computed by subtracting the header length from this value. The size of the total datagram length field is 16 bits, hence the 65,535 bytes are the maximum length of a datagram (including the header). This field is used to determine the length value to be passed to the transport protocol to set the total packet length.

Identification

This field holds a number that is a unique identifier created by the sending node. This number is required when reassembling fragmented messages, ensuring that the fragments of one message are not intermixed with others. Each chunk of data received by the IP layer from a higher protocol layer is assigned one of these identification numbers when the data arrives. If a datagram is fragmented, each fragment has the same identification number.

Flags

The Flags field is a 3-bit field, the first bit of which is left unused (it is ignored by the protocol and usually has no value written to it). The remaining two bits are dedicated to flags called DF (Don't Fragment) and MF (More Fragments), which control the handling of the datagrams when fragmentation is desirable.

If the DF flag is set to 1, the datagram cannot be fragmented under any circumstances. If the current IP layer software cannot send the datagram to another machine without fragmenting it, and this bit is set to 1, the datagram is discarded and an error message is sent back to the sending device.

If the MF flag is set to 1, the current datagram is followed by more fragments (sometimes called *subpackets*), which must be reassembled to re-create the full message. The last fragment that is sent as part of a larger message has its MF flag set to 0 (off) so that the receiving device knows when to stop waiting for datagrams. Because the order of the fragments' arrival might not correspond to the order in which they were sent, the MF flag is used in conjunction with the Fragment Offset field (the next field in the IP header) to indicate to the receiving machine the full extent of the message.

Fragment Offset

If the MF (More Fragments) flag bit is set to 1 (indicating fragmentation of a larger datagram), the fragment offset contains the position of the fragments contained in the complete message within the current datagram. This enables IP to reassemble fragmented packets in the proper order.

Offsets are always given relative to the beginning of the message. This is a 13-bit field, so offsets are calculated in units of 8 bytes, corresponding to the maximum packet length of 65,535 bytes. Using the identification number to indicate which message a receiving datagram belongs to, the IP layer on a receiving machine can then use the fragment offset to reassemble the entire message.

Time to Live (TTL)

This field gives the amount of time in maximum number of **hops** that a datagram can remain on the network before it is discarded. This is set by the sending node when the datagram is assembled. Usually the TTL field is set to 15 or 30 hops.

The TCP/IP standards stipulate that the TTL field must be decreased by at least one hop for each node that processes the packet, even if the processing time is less. Also, when a datagram is received by a gateway, the arrival time is tagged so that if the datagram must wait to be processed, that time counts against its TTL. Hence, if a gateway is particularly overloaded and can't get to the datagram in short order, the TTL timer can expire while awaiting processing, and the datagram is abandoned.

If the TTL field reaches 0, the datagram must be discarded by the current node, but a message is sent back to the sending machine when the packet is dropped. The sending machine can then resend the datagram. The rules governing the TTL field are designed to prevent IP packets from endlessly circulating through networks.

Protocol

This field holds the identification number of the protocol to which the packet is to be handed. The numbers are defined by the Network Information Centre (NIC), which governs the Internet. There are currently about 50 protocols defined and assigned a transport protocol number. The two most important protocols are ICMP (detailed in the section titled “Internet Control Message Protocol (ICMP)” later today), which is number 1, and TCP, which is number 6. The full list of numbers is not necessary here because most of the protocols are never encountered by users.

Header Checksum

The number in this field of the IP header is a checksum for the protocol header field (but not the data fields) to enable faster processing. Because the Time to Live (TTL) field is decremented at each node, the checksum also changes with every machine the datagram passes through. The checksum algorithm takes the ones-complement of the 16-bit sum of all 16-bit words. This is a fast, efficient algorithm, but it misses some unusual corruption circumstances such as the loss of an entire 16-bit word that contains only 0s. However, because the data checksums used by both TCP and UDP cover the entire packet, these types of errors usually can be caught as the packet is assembled for the network transport.

Sending Address and Destination Address

These fields contain the 32-bit IP addresses of the sending and destination devices. These fields are established when the datagram is created and are not altered during the routing.

Options

The Options field is optional, composed of several codes of variable length. If more than one option is used in the datagram, the options appear consecutively in the IP header. All the options are controlled by a byte that is usually divided into three fields: a 1-bit copy flag, a 2-bit option class, and a 5-bit option number. The copy flag is used to stipulate how the option is handled when fragmentation is necessary in a gateway. When the bit is set to 0, the option should be copied to the first datagram but not subsequent ones. If the bit is set to 1, the option is copied to all the datagrams.

The option class and option number indicate the type of option and its particular value. At present, there are only two option classes set. (With only 2 bits to work with in the field, a maximum of four options could be set.) When the value is 0, the option applies to datagram or network control. A value of 2 means the option is for debugging or administration purposes. Values of 1 and 3 are unused. Currently supported values for the option class and number are given in *Table 1*.

Table 1: Valid option class and numbers for IP headers

Option Class	Option Number	Description
0	0	Marks the end of the options list
0	1	No option (used for padding)
0	2	Security options (military purposes only)
0	3	Loose source routing
0	7	Activates routing record (adds fields)
0	9	Strict source routing
2	4	Timestamping active (adds fields)

Of most interest to you are options that enable the routing and timestamps to be recorded. These are used to provide a record of a datagram's passage across the internetwork, which can be useful for diagnostic purposes. Both these options add information to a list contained within the datagram. (The timestamp has an interesting format: it is expressed in milliseconds since midnight, Universal Time. Unfortunately, because most systems have widely differing time settings — even when corrected to Universal Time — the timestamps should be treated with more than a little suspicion).

There are two kinds of routing indicated within the options field: loose and strict. *Loose routing* provides a series of IP addresses that the machine must pass through, but it enables any route to be used to get to each of these addresses (usually gateways). *Strict routing* enables no deviations from the specified route. If the route can't be followed, the datagram is abandoned. Strict routing is frequently used for testing routes but rarely for transmission of user datagrams because of the higher chances of the datagram being lost or abandoned.

Padding

The content of the padding area depends on the options selected. The padding is usually used to ensure that the datagram header is a round number of bytes.

2.4 IP ADDRESSES

The Internet Protocol moves data between hosts in the form of datagrams. Each datagram is delivered to the address contained in the Destination Address (word 5) of the datagram's header. The Destination Address is a standard 32-bit IP address that contains sufficient information to uniquely identify a network and a specific host on that network.

An IP address contains a network part and a host part, but the format of these parts is not the same in every IP address. The number of address bits used to identify the network, and the number used to identify the host, vary according to the prefix length of the address. There are two ways the prefix length is determined: by address class or by a CIDR address mask. We begin with a discussion of traditional IP address classes.

2.4.1 IP Address Classes

Originally, the IP address space was divided into a few fixed-length structures called address classes. The three main address classes are class A, class B, and class C. By examining the first few bits of an address, IP software can quickly determine the class, and therefore the structure, of an address. IP follows these rules to determine the address class:

- **Class A:** If the first bit of an IP address is 0, it is the address of a class A network. The first bit of a class A address identifies the address class. The next 7 bits identify the network, and the last 24 bits identify the host. There are fewer than 128 class A network numbers, but each class A network can be composed of millions of hosts.

- **Class B:** If the first 2 bits of the address are 1 0, it is a class B network address. The first 2 bits identify class; the next 14 bits identify the network, and the last 16 bits identify the host. There are thousands of class B network numbers and each class B network can contain thousands of hosts.

- **Class C:** If the first 3 bits of the address are 1 1 0, it is a class C network address. In a class C address, the first 3 bits are class identifiers; the next 21 bits are the network address, and the last 8 bits identify the host. There are millions of class C network numbers, but each class C network is composed of fewer than 254 hosts.

- **Class D:** If the first 4 bits of the address are 1 1 1 0, it is a multicast address. These addresses are sometimes called class D addresses, but they don't really refer to specific networks. Multicast addresses are used to address groups of computers all at one time. Multicast addresses identify a group of computers that share a common application, such as a video conference, as opposed to a group of computers that share a common network.

- **Class E:** If the first four bits of the address are 1 1 1 1, it is a special reserved address. These addresses are called class E addresses, but they don't really refer to specific networks. No numbers are currently assigned in this range.

IP addresses are usually written as four decimal numbers separated by dots (periods). Each of the four numbers is in the range 0-255 (the decimal values possible for a single byte). Because the bits that identify class are contiguous with the network bits of the address, we can lump them together and look at the address as composed of full bytes of network address and full bytes of host address. If the value of the first byte is:

- Less than 128, the address is class A; the first byte is the network number, and the next three bytes are the host address.
- From 128 to 191, the address is class B; the first two bytes identify the network, and the last two bytes identify the host.
- From 192 to 223, the address is class C; the first three bytes are the network address, and the last byte is the host number.
- From 224 to 239, the address is multicast. There is no network part. The entire address identifies a specific multicast group.
- Greater than 239, the address is reserved.

The Table 2 depicts each class range with other details.

Table 2: IP Address Classes in dotted decimal format with their ranges

IP Address Class	High Order Bit(s)	Format	Range	No. of Network Bits	No. of Host Bits	Max. Hosts	Purpose
A	0	N.H.H.H	1.0.0.0 to 126.0.0.0	7	24	2^{24}-2	Few large organisations
B	1,0	N.N.H.H	128.1.0.0 to 191.254.0.0	14	16	2^{16}-2	Medium-size organisations
C	1,1,0	N.N.N.H	192.0.1.0 to 223.255.254.0	21	8	2^{8}-2	Relatively small organisations
D	1,1,1,0	N/A	224.0.0.0 to 239.255.255.255	N/A	N/A	N/A	Multicast groups (RFC 1112)
E	1,1,1,1	N/A	240.0.0.0 to 254.255.255.255	N/A	N/A	N/A	Future Use (Experimental)

The IP address, which provides universal addressing across all of the networks of the Internet, is one of the great strengths of the TCP/IP protocol suite. However, the original class structure of the IP address has weaknesses. The TCP/IP designers did not envision the enormous scale of today's network. When TCP/IP was being

designed, networking was limited to large organisations that could afford substantial computer systems. The idea of a powerful UNIX system on every desktop did not exist. At that time, a 32-bit address seemed so large that it was divided into classes to reduce the processing load on routers, even though dividing the address into classes sharply reduced the number of host addresses actually available for use. For example, assigning a large network a single class B address, instead of six class C addresses, reduced the load on the router because the router needed to keep only one route for that entire organisation. However, an organisation that was given the class B address probably did not have 64,000 computers, so most of the host addresses available to the organisation were never assigned.

The class-structured address design was critically strained by the rapid growth of the Internet. At one point it appeared that all class B addresses might be rapidly exhausted. To prevent this, a new way of looking at IP addresses without a class structure was developed.

Check Your Progress 1

1) Internet Protocol is a _____ layer protocol.

   a) Transport

   b) Application

   c) Network

   d) Data Link

   ……

2) What is the size of “type of service” field?

   a) 4 bit

   b) 5 bit

   c) 7 bit

   d) 8 bit

   ……

3) If DF flag is set to 1, it means the datagram:

   a) Large and need fragmentation

   b) Small & need reassembling

   c) Cannot be fragmented

   d) Is accepted and message is sent back to sending device

   ……

2.4.2 Subnet Masks and CIDR Networks (Classless IP Addresses)

IP addresses are actually 32-bit binary numbers. Each 32-bit IP address consists of two subaddresses, one identifying the network and the other identifying the host to the network, with an imaginary boundary separating the two. The location of the boundary between the network and host portions of an IP address is determined through the use of a subnet mask. A subnet mask is another 32-bit binary number, which acts like a filter when it is applied to the 32-bit IP address. By comparing a subnet mask with an IP address, systems can determine which portion of the IP address relates to the network, and which portion relates to the host. Anywhere the subnet mask has a bit set to "1", the underlying bit in the IP address is part of the network address. Anywhere the subnet mask is set to "0", the related bit in the IP

address is part of the host address. For example, assume that the IP address 11000000101010000000000100010100 has a subnet mask of 11111111111111111111111100000000. In this example, the first 24 bits of the 32-bit IP addresses are used to identify the network, while the last 8 bits are used to identify the host on that network.

The size of a network (i.e., the number of host addresses available for use on it) is a function of the number of bits used to identify the host portion of the address. If a subnet mask shows that 8 bits are used for the host portion of the address block, a maximum of 256 possible host addresses are available for that specific network. Similarly, if a subnet mask shows that 16 bits are used for the host portion of the address block, a maximum of 65,536 possible host addresses are available for use on that network.

If a network administrator needs to split a single network into multiple virtual networks, the bit-pattern in use with the subnet mask can be changed to allow as many networks as necessary. For example, assume that we want to split the 24-bit 192.168.10.0 network (which allows for 8 bits of host addressing, or a maximum of 256 host addresses) into two smaller networks. All we have to do in this situation is to change the subnet mask of the devices on the network so that they use 25 bits for the network instead of 24 bits, resulting in two distinct networks with 128 possible host addresses on each network. In this case, the first network would have a range of network addresses between 192.168.10.0 -192.168.10.127, while the second network would have a range of addresses between 192.168.10.128 -192.168.10.255.

Networks can also be enlarged through the use of a technique known as "**supernetting**," which works by extending the host portion of a subnet mask to the left, into the network portion of the address. Using this technique, a pair of networks with 24-bit subnet masks can be turned into a single large network with a 23-bit subnet mask. However, this works only if you have two neighbouring 24-bit network blocks, with the lower network having an even value (when the network portion of the address is shrunk, the trailing bit from the original network portion of the subnet mask will fall into the host portion of the new subnet mask, so the new network mask will consume both networks). For example, it is possible to combine the 24-bit 192.168.10.0 and 192.168.11.0 networks together since the loss of the trailing bit from each network (00001010 vs. 00001011) produces the same 23-bit subnet mask (0000101x), resulting in a consolidated 192.168.10.0 network. However, it is not possible to combine the 24-bit 192.168.11.0 and 192.168.12.0 networks, since the binary values in the seventh bit position (00001011 vs. 00001100) do not match when the trailing bit is removed.

Classless Inter-Domain Routing

In the modern networking environment defined by RFC 1519 [Classless Inter-Domain Routing (CIDR)], the subnet mask of a network is typically annotated in written form as a "slash prefix" that trails the network number. In the subnetting example in the previous paragraph, the original 24-bit network would be written as 192.168.10.0/24, while the two new networks would be written as 192.168.10.0/25 and 192.168.10.128/25. Likewise, when the 192.168.10.0/24 and 192.168.11.0/24 networks were joined together as a single supernet, the resulting network would be written as 192.168.10.0/23. Note that the slash prefix annotation is generally used for human benefit; infrastructure devices still use the 32-bit binary subnet mask internally to identify networks and their routes. All networks must reserve host addresses that are made up entirely of either ones or zeros, to be used by the networks themselves. This is so that each subnet will have a network-specific address (the all-zeroes address) and a broadcast address (the all-ones address). For example, a /24 network allows for 8 bits of host addresses, but only 254 of the 256 possible addresses are available for use. Similarly, /25 networks have a maximum of 7 bits for host addresses, with 126 of the 128 possible addresses available (the all-ones and all-

zeroes addresses from each subnet must be set aside for the subnets themselves). All the systems on the same subnet must use the same subnet mask in order to communicate with each other directly. If they use different subnet masks they will think they are on different networks, and will not be able to communicate with each other without going through a router first. Hosts on different networks can use different subnet maks, although the routers will have to be aware of the subnet masks in use on each of the segments.

Subnet masks are used only by systems that need to communicate with the network directly. For example, external systems do not need to be aware of the subnet masks in use on your internal networks, since those systems will route data to your networks by way of your parent network's address block. As such, remote routers need to know only about your provider's subnet mask. For example, if you have a small network that uses only a /28 prefix that is a subset of your ISP's /20 network, remote routers need to know only about your upstream provider's /20 network, while your upstream provider needs to know your subnet mask in order to get the data to your local /28 network. The rapid depletion of the class B addresses showed that three primary address classes were not enough: class A was much too large and class C was much too small. Even a class B address was too large for many networks but was used because it was better than the alternatives.

The obvious solution to the class B address crisis was to force organisations to use multiple class C addresses. There were millions of these addresses available and they were in no immediate danger of depletion. As is often the case, the obvious solution is not as simple as it may seem. Each class C address requires its own entry within the routing table. Assigning thousands or millions of class C addresses would cause the routing table to grow so rapidly that the routers would soon be overwhelmed. The solution require a new way of assigning addresses and a new way of looking at addresses.

Originally network addresses were assigned in more or less sequential order as they were requested. This worked fine when the network was small and centralised. However, it did not take network topology into account. Thus only random chance would determine if the same intermediate routers would be used to reach network 195.4.12.0 and network 195.4.13.0, which makes it difficult to reduce the size of the routing table. Addresses can only be aggregated if they are contiguous numbers and are reachable through the same route. For example, if addresses are contiguous for one service provider, a single route can be created for that aggregation because that service provider will have a limited number of routes to the Internet. But if one network address is in France and the next contiguous address is in Australia, creating a consolidated route for these addresses does not work.

Today, large, contiguous blocks of addresses are assigned to large network service providers in a manner that better reflects the topology of the network. The service providers then allocate chunks of these address blocks to the organisations to which they provide network services. This alleviates the short-term shortage of class B addresses and, because the assignment of addressees reflects the topology of the network, it permits route aggregation. Under this new scheme, we know that network 195.4.12.0 and network 195.4.13.0 are reachable through the same intermediate routers. In fact, both of these addresses are in the range of the addresses assigned to Europe, 194.0.0.0 to 195.255.255.255. Assigning addresses that reflect the topology of the network enables route aggregation, but does not implement it. As long as network 195.4.12.0 and network 195.4.13.0 are interpreted as separate class C addresses, they will require separate entries in the routing table. A new, flexible way of defining addresses is needed.

Evaluating addresses according to the class rules discussed above limits the length of network numbers to 8, 16, or 24 bits - 1, 2, or 3 bytes. The IP address, however, is not really byte-oriented. It is 32 contiguous bits. A more flexible way to interpret the

network and host portions of an address is with a bit mask. An address bit mask works in this way: if a bit is on in the mask, that equivalent bit in the address is interpreted as a network bit; if a bit in the mask is off, the bit belongs to the host part of the address. For example, if address 195.4.12.0 is interpreted as a class C address, the first 24 bits are the network numbers and the last 8 bits are the host addresses. The network mask that represents this is 255.255.255.0, 24 bits on and 8 bits off. The bit mask that is derived from the traditional class structure is called the default mask or the natural mask.

However, with bit masks we are no longer limited by the address class structure. A mask of 255.255.0.0 can be applied to network address 195.4.0.0. This mask includes all addresses from 195.4.0.0 to 195.4.255.255 in a single network number. In effect, it creates a network number as large as a class B network in the class C address space. Using bit masks to create networks larger than the natural mask is called supernetting, and the use of a mask instead of the address class to determine the destination network is called Classless Inter-Domain Routing (CIDR).

Specifying both the address and the mask is cumbersome when writing out addresses. A shorthand notation has been developed for writing CIDR addresses. Instead of writing network 172.16.26.32 with a mask of 255.255.255.224, we can write 172.16.26.32/27. The format of this notation is address/prefix-length, where prefix-length is the number of bits in the network portion of the address. Without this notation, the address 172.16.26.32 could easily be interpreted as a host address. RFC 1878 list all 32 possible prefix values. But little documentation is needed because the CIDR prefix is much easier to understand and remember than are address classes. I know that 10.104.0.19 is a class A address, but writing it as 10.104.0.19/8 shows me that this address has 8 bits for the network number and therefore 24 bits for the host number. I don't have to remember anything about the class A address structure.

2.4.3 Internet-Legal Versus Private Addressing

Although the pool of IP addresses is somewhat limited, most companies have no problems obtaining them. However, many organisations have already installed TCP/IP products on their internal networks without obtaining "legal" addresses from the proper sources. Sometimes these addresses come from example books or are simply picked at random (several firms use networks numbered 1.2.3.0, for example). Unfortunately, since they are not legal, these addresses will not be usable when these organisations attempt to connect to the Internet. These firms will eventually have to reassign Internet-legal IP addresses to all the devices on their networks, or invest in address-translation gateways that rewrite outbound IP packets so they appear to be coming from an Internet-accessible host.

Even if an address-translation gateway is installed on the network, these firms will never be able to communicate with any site that is a registered owner of the IP addresses in use on the local network. For example, if you choose to use the 36.0.0.0/8 address block on your internal network, your users will never be able to access the computers at Stanford University, the registered owner of that address block. Any attempt to connect to a host at 36.x.x.x will be interpreted by the local routers as a request for a local system, so those packets will never leave your local network.

Not all firms have the luxury of using Internet-legal addresses on their hosts, for any number of reasons. For example, there may be legacy applications that use hardcode addresses, or there may be too many systems across the organisation for a clean upgrade to be successful. If you are unable to use Internet-legal addresses, you should at least be aware that there are groups of "private" Internet addresses that can be used on internal networks by anyone. These address pools were set-aside in RFC 1918, and therefore cannot be "assigned" to any organization. The Internet's backbone routers are configured explicitly not to route packets with these addresses, so they are

completely useless outside an organization's internal network. The address blocks available are listed in *Table 3*.

Table 3: Private Addresses Provided in RFC 1918

Class	Range of Addresses
A	Any addresses in 10.x.x.x
B	Addresses in the range of 172.16.x.x-172.31.x.x
C	Addresses in the range of 192.168.0.x-192.168.255.x

Since these addresses cannot be routed across the Internet, you must use an address-translation gateway or a proxy server in conjunction with them. Otherwise, you will not be able to communicate with any hosts on the Internet.

An important note here is that since nobody can use these addresses on the Internet, it is safe to assume that anybody who is using these addresses is also utilising an address-translation gateway of some sort. Therefore, while you will never see these addresses used as destinations on the Internet, if your organisation establishes a private connection to a partner organisation that is using the same block of addresses that you are using, your firms will not be able to communicate. The packets destined for your partner's network will appear to be local to your network, and will never be forwarded to the remote network.

There are many other problems that arise from using these addresses, making their general usage difficult for normal operations. For example, many application-layer protocols embed addressing information directly into the protocol stream, and in order for these protocols to work properly, the address-translation gateway has to be aware of their mechanics. In the preceding scenario, the gateway has to rewrite the private addresses (which are stored as application data inside the application protocol), rewrite the UDP/TCP and IP checksums, and possibly rewrite TCP sequence numbers as well. This is difficult to do even with simple and open protocols such as FTP, and extremely difficult with proprietary, encrypted, or dynamic applications (these are problems for many database protocols, network games, and voice-over-IP services, in particular). These gateways almost never work for all the applications in use at a specific location.

It is always best to use formally-assigned, Internet-legal addresses whenever possible, even if the hosts on your network do not necessarily require direct Internet access. In those cases in which your hosts are going through a firewall or application proxy of some sort, the use of Internet-legal addresses causes the least amount of maintenance trouble over time. If for some reason this is not possible, use one of the private address pools described in *Table 3*. Do not use random, self-assigned addresses if you can possibly avoid it, as this will only cause connectivity problems for you and your users.

☞ Check Your Progress 2

1) If the first bit of an IP address is zero, then it belongs to

   a) Class A

   b) Class B

   c) Class C

   d) Class D

   ……

2) The Internet backbone renters are configured explicitly not to route packets with _____ address.

   a) Legal

   b) Private

   c) Public

   d) Virtual

   ……

3) _____ Works by extending the host portion of a subnet mask to the left, into the network portion of the address.

   a) Subnetting

   b) Supernetting

   c) Classless addressing

   d) Hyper testing

   ……

2.5 IP ROUTING

An important function of the IP layer is *IP routing*. It provides the basic mechanism for routers to interconnect different physical networks. This means that an internet host can function as a normal host and a router simultaneously.

A basic router of this type is referred to as a *router with partial routing information*, because the router only has information about four kinds of destination:

- Hosts which are directly attached to one of the physical networks to which the router is attached.
- Hosts or networks for which the router has been given explicit definitions.
- Hosts or networks for which the router has received an ICMP redirect message.
- A default destination for everything else.

The last two items allow a basic router to begin with a very limited amount of information and to increase its information because a more sophisticated router will issue an ICMP redirect message if it receives a datagram and it knows of a better router on the same network for the sender to use. This process is repeated each time a basic router of this type is restarted.

Additional protocols are needed to implement a full-function router that can exchange information with other routers in remote network. Such routers are essential except in small networks, and the protocols they use are discussed in Routing Protocol.

2.5.1 Routing Protocol

The Routing protocols evolved in different phases. There are some references are as below:

- RFC 1074 - The NSFNET Backbone SPF Based Interior Gateway Protocol.
- RFC 1092 - EGP and Policy Based Routing in the New NSFNET Backbone.
- RFC 1093 - The NSFNET Routing Architecture.
- RFC 1104 - Models of Policy Based Routing.

- RFC 1133 - Routing between the NSFNET and the DDN.
- RFC 1222 - Advancing the NSFNET Routing Architecture

These protocols use two important groups of routing algorithms.

2.5.2 Routing Algorithms

In this section, we discuss the Distance-Vector and Link-State, Shortest Path First (SPF) Routing Algorithms.

Distance-Vector

The term *Distance-Vector* refers to a class of algorithms that gateways use to update routing information. Each router begins with a set of routes for those networks or subnets to which it is directly attached, and possibly some additional routes to other networks or hosts if the network topology is such that the routing protocol will be unable to produce the desired routing correctly. This list is kept in a *routing table*, where each entry identifies a destination network or host and gives the "distance"' to that network. The distance is called a *metric* and is typically measured in "hops".

Periodically, each router sends a copy of its routing table to any other router it can reach directly. When a report arrives at router B from router A, B examines the set of destinations it receives and the distance to each. B will update its routing table if:

- A knows a shorter way to reach a destination.
- A lists a destination that B does not have in its table.
- A's distance to a destination, already routed through A from B, has changed.

This kind of algorithm is easy to implement, but it has a number of disadvantages:

- When routes change rapidly, that is, a new connection appears or an old one fails, the routing topology may not stabilize to match the changed network topology because information propagates slowly from one router to another and while it is propagating, some routers will have incorrect routing information.

- Another disadvantage is that each router has to send a copy of its entire routing table to every neighbour at regular intervals. Of course, one can use longer intervals to reduce the network load but that introduces problems related to how well the network responds to changes in topology.

- Vector-distance algorithms using hop counts as a metric does not take account of the link speed or reliability. Such an algorithm will use a path with hop count 2 that crosses two slow-speed lines, instead of using a path with hop count 3 that crosses three token-rings and may be substantially faster.

The most difficult task in a distance vector algorithm is to prevent instability. Different solutions are available:

Counting to infinity

Let us choose a value of 16 to represent infinity. Suppose a network becomes inaccessible, all the immediately neighbouring routers time out and set the metric to that network to 16. We can consider that all the neighbouring routers have a piece of hardware that connects them to the vanished network, with a cost of 16. Since that is the only connection to the vanished network, all the other routers in the system will converge to new routes that go through one of those routers with a direct but unavailable connection. Once convergence has happened, all the routers will have metrics of 16 for the vanished network. Since 16 indicates infinity, all routers then regard the network as unreachable.

The growth in networking over the past few years has pushed the currently available *Interior Gateway Protocols*, which use distance-vector algorithms, past their limits.

The primary alternative to vector-distance schemes is a class of protocols known as *Link State, Shortest Path First*.

The important features of these routing protocols are:

- A set of physical networks is divided into a number of areas.
- All routers within an area have an identical database.
- Each router's database describes the complete topology (which routers are connected to which networks) of the routing domain. The topology of an area is represented with a database called a *Link State Database* describing all of the links that each of the routers in the area has.
- Each router uses its database to derive the set of optimum paths to all destinations from which it builds its routing table. The algorithm used to determine the optimum paths is called a *Shortest Path First (SPF)* algorithm.

In general, a link state protocol works as follows each router periodically sends out a description of its connections (the state of its links) to its neighbours (routers are neighbours if they are connected to the same network). This description, called a *Link State Advertisement (LSA)*, includes the configured cost of the connection. The LSA is flooded throughout the router's domain. Each router in the domain maintains an identical synchronised copy of a database composed of this link state information. This database describes both the topology of the router's domain and routes to networks outside of the domain such as routes to networks in other autonomous systems. Each router runs an algorithm on its topological database resulting in a shortest-path tree. This shortest-path tree contains the shortest path to every router and network the gateway can reach. From the shortest-path tree, the cost to the destination and the next hop to forward a datagram is used to build the router's routing table.

Link-state protocols, in comparison with vector-distance protocols, send out updates when there is news, and may send out regular updates as a way of ensuring neighbour routers that a connection is still active. More importantly, the information exchanged is the state of a router's links, not the contents of the routing table. This means that link-state algorithms use fewer network resources than their vector-distance counterparts, particularly when the routing is complex or the autonomous system is large. They are, however, compute-intensive. In return, users get faster response to network events, faster route convergence, and access to more advanced network services.

☞ Check Your Progress 3

1) Which of the following device provides the basic mechanism for interconnect different physical networks.

   a) Gateways

   b) Bridges

   c) Routers

   d) Hubs

   ……

2) _____ routing algorithm all routers within an area have an identical database.

   a) Distance-vector

   b) Link state

   c) Longest-path

   d) Shortest path

   ……

3) _____ Routing algorithms use features network resources than the distance-vector routing.

   a) Link-state

   b) Policy based

   c) Connection based

   d) Hops based

   ……

2.6 SUMMARY

In this unit, you have studied the architecture of the Internet protocol, you must have learned different fields of IP header and their role in network communication. You learned about the different IP classes and their use in network design. You also studied about the practical issues of subnettting, supernetting and the Classless IP Addresses. Further in this unit we have given you comparisons between Internet-legal addressing and private addressing. In the end of this unit we have explained the procedure of IP routing with the help of Distance-Vector and Link-State routing algorithms. The next unit of this course covers about the transport layer and its role in TCP/IP.

2.7 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) C 2) D 3) C

Check Your Progress 2

1) A 2) B 3) B

Check Your Progress 3

1) C 2) B 3) A

2.8 FURTHER READINGS

1) Achyut S. Godbole, *Web Technoligies* TATA McGrawHill, 2003.

2) Berhouz Forouzan, *TCP/IP Protocol Suite*, 3rd edition, TATA McGraw Hill, 2006.

3) http://www.tcpipguide.com.

4) http://www.cisco.com.

UNIT 3 TRANSPORT LAYER PROTOCOLS

Structure **Page Nos.**

3.0 Introduction 61

3.1 Objectives 61

3.2 Overview of TCP 62

3.3 Transmission Control Protocol (TCP) 63

3.3.1 TCP Header 64

3.3.2 TCP Connection Establishment and Termination 67

3.3.3 TCP Connection Establishment 67

3.3.4 TCP Connection Termination 68

3.4 User Datagram Protocol (UDP) 72

3.5 Summary 74

3.6 Answers to Check Your Progress 75

3.7 Further Readings 75

3.0 INTRODUCTION

The internet work environment consists of hosts connected to networks that are in turn interconnected via gateways. Such networks are based on packet switching technology. The active elements that produce and consume messages are the *processes*. Therefore, such communication can be viewed as an inter-process communication.

As the computer communication systems are playing an important role in military, government and civilian environments, it has become essential to provide some means of interconnecting the communication networks and to provide standard inter-process communication protocols that can support a broad range of applications. In anticipation of the need for such standards, the Department of Defence (DoD), USA, decided to employ Transmission Control Protocol (TCP) as the basis of inter-process communication protocol.

3.1 OBJECTIVES

Our objective is to introduce you to the basic concepts of Transport layer Protocols. On successful completion of this unit, you should be able to:

- have a reasonable understanding of the Transmission Control Protocol architecture;
- understand the role and use of port numbers;
- describe the operation of Transmission Control Protocol and its header format;
- describe the role and meaning of User Datagram Protocol (UDP); and
- describe the role and meaning of Internet Control Message Protocol (ICMP).

Ports

When a process starts up, it registers a port number with the protocol stack. The port numbers are specified by a 16-bit number i.e., the overall available set of port numbers are 2^{16}, ranging from 0 to 65535. The various categories of ports are as under:

1) **Well Known Ports:** The Port numbers in the range 0-1023 are called 'Well Known Ports'. These port numbers are assigned to the server side of an application and are already reserved for specific applications by IANA (Internet Assigned Number Authority). Some of these port numbers and their associated application are given in *Table 1*.

Table 1: Port number and associated application

Port Nos.	Protocol	Application	Port No	Protocol	Application
23	TCP	TELNET	**110**	TCP	POP3
25	TCP	SMTP	**161**	UDP	SNMP
37	UDP	TIME	**179**	TCP	BGP
43	TCP	WHOIS	**443**	TCP	HTTPS
53	TCP/UDP	DNS	**68**	UDP	BOOTPC
67	UDP	BOOTPS	**69**	UDP	TFTP

2) **Registered Ports:** Port numbers in the range 1024-49151 are called Registered Ports. These port numbers have been publicly defined as a convenient service for the Internet community to help them avoid vendor conflicts.

3) **Dynamic and/or Private Ports:** The remaining port numbers in the range 49152-65535, are called Dynamic and/or Private Ports and can be used freely by any client or server application.

Remember, only one process per protocol can listen on a given port i.e., two different processes, one using UDP and another TCP can both listen on port number XXX. However, two processes using the same transport protocol cannot listen on the same port number.

In order to establish a connection between two end machines i.e., providing the process-to-process delivery of messages, we need an identifier called socket. It contains the combination of IP address of the machine along with the port number of the process running on it. For example, suppose IP address of the machine is 171.10.20.1 and a process running on it is assigned a port number of 4534, then the socket contains (171.10.20.1, 4544). Thus, a socket uniquely identifies the process from the set of processes running on a machine.

In the client server model, a pair of sockets is required i.e., one for the client *(client IP address, client port number)* and another for the server *(server IP address, server port number)*.

3.2 OVERVIEW OF TCP

In the previous discussion, we have discussed that in data link layer, frames are delivered node-to-node (intermediate nodes) and in the network layer, datagrams are delivered host-to-host. But the real communication takes place between two processes running on two different machines. Therefore, the basic function of the transport layer is process-to-process delivery of messages. However, it is also necessary to identify the various kinds of processes running on the machine (e.g., e-mail, ftp, telnet). Thus, we need some kind of addressing scheme for transfer of data between the processes.

In order to deliver some information to a specific destination, we need an address for correspondence. Similarly, at the data link layer, medium access control addresses (MAC) are employed for addressing. Correspondingly, at the network layer, Internet Protocol IP addresses are required. In the same way, the transport layer uses the port numbers for addressing between two processes. These port numbers are just a way of labelling the packets. When the transport layer receives a message from an application, first of all, it registers a temporary port number for the application process and subsequently assigns it. Thereafter, it sends the data to the designated destination application program that would be running on another port number. In *Figure 1*, the client server model has been illustrated, wherein, whenever a client process starts up, a port number is assigned, say 3000 and the server process is assigned port number of 100. These assigned port numbers are put on each packet transmitted between these processes.

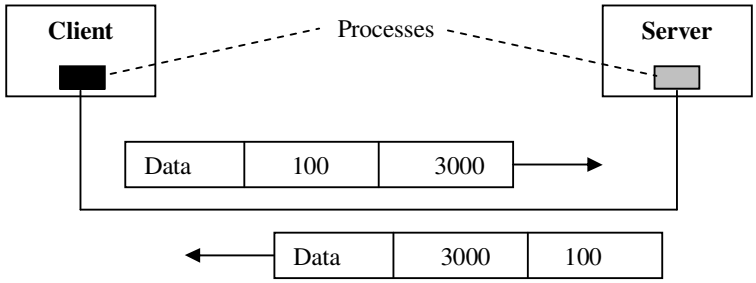


Figure: 1 Client Server Model

It may be noted that IP addresses and port numbers play different roles in communication networks. The IP address simply identifies the destination machine from the network, while the port number identifies the particular process running on that destination machine.

There are two types of protocols at the transport layer as shown in *Figure.2*.

1) Transmission Control Protocol (TCP)

2) User Datagram Protocol (UDP).

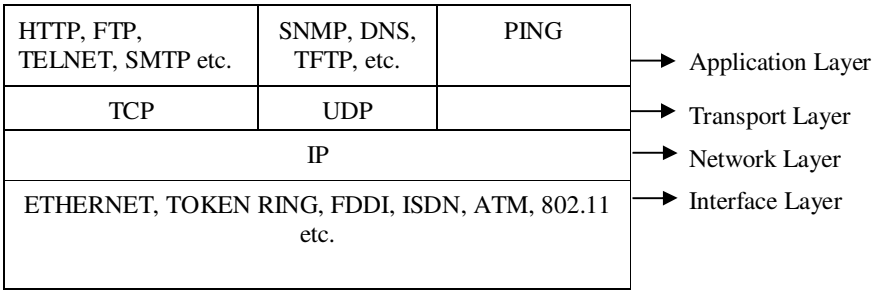


Figure 2: Protocol Stack

3.3 TRANSMISSION CONTROL PROTOCOL (TCP)

Transmission control protocol (TCP) of the transport layer provides end-to-end reliable communication. TCP is a connection oriented communication service. As the IP layer of TCP/IP protocol stack is connectionless, the TCP layer adds the reliability feature along with connection-oriented service. Basically, TCP establishes a virtual circuit between the source and the destination and the stream of bytes are transferred through that circuit such that the order of the segments remains integral.

The acknowledgement and retransmission of lost and damaged segments is guaranteed. In TCP, two processes running on two different hosts communicate with each other as shown in *Figure3*.

The various services being offered by TCP to the processes running on application layer are as under:

1) **Stream data transfer**: With stream data transfer**,** TCP delivers an unstructured stream of bytes identified by sequence numbers. This service benefits applications because they do not have to chop data into blocks before handing it off to TCP. Instead, TCP groups bytes into segments and passes them to IP for delivery.

2) **Reliability**: TCP offers reliability by providing connection-oriented, end-to-end reliable packet delivery through an internetwork. It does this by sequencing bytes with a forwarding acknowledgement number that indicates to the destination, the next byte the source expects to receive. Bytes not acknowledged within a specified time period are retransmitted. The reliability mechanism of TCP allows devices to deal with lost, delayed, duplicate or misread packets. A time-out mechanism allows devices to detect lost packets and request retransmission.

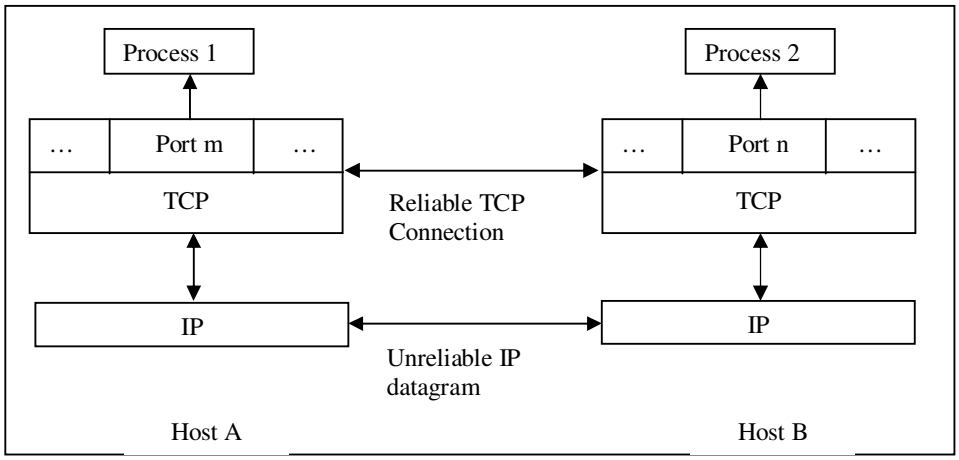


Figure 3: TCP Connection between two processes running on different machines

3) **Efficient flow control**: TCP offers efficient flow control, which means that, when sending acknowledgements back to the source, the receiving TCP process indicates the highest sequence number it can receive without overflowing its internal buffers.

4) **Full-duplex operation**: TCP offers full-duplex operation, wherein TCP processes can both send and receive data at the same time.

5) Multiplexing: TCP's multiplexing provides numerous simultaneous upper-layer conversations multiplexed over a single connection.

3.3.1 TCP Header

The TCP data unit is called a *segment*. The format of the segment is shown in *Figure 4*.

<table>
<tr><td colspan="9">Source Port address
16 bits</td><td colspan="8">Destination Port address
16 bits</td></tr>
<tr><td colspan="17">Sequence Number
32 bits</td></tr>
<tr><td colspan="17">Acknowledgement Number
32 bits</td></tr>
<tr><td colspan="4">HLEN
4 bits</td><td colspan="4">Reserved
6 bits</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
H</td><td>F
I
N</td><td colspan="8">Window Size
16 bits</td></tr>
<tr><td colspan="9">Checksum
16 bits</td><td colspan="8">Urgent Pointer
16 bits</td></tr>
<tr><td colspan="9">Options</td><td colspan="8">Padding</td></tr>
</table>

Figure 4: TCP Header

1) **Source Port Number**: This is a 16-bit number which defines the source port number for a particular application program that is sending the TCP segments.

2) **Destination Port Number**: This is a 16-bit number which defines the destination port number for a particular application program that is receiving the TCP segments.

3) **Sequence Number**: As the unit of data transfer in TCP is termed as segment, each segment's first data byte number denotes the 32-bit sequence number. Since the sequence number refers to a byte count rather than a segment count, sequence numbers in contiguous TCP segments are not numbered sequentially. For example, if a file of 5000 bytes is transferred using TCP connection and the first byte is numbered as 20002 and data divided into 5 segments each of 1000 bytes, the sequence numbers assigned to various segments are as under:

   Segment 1 → sequence number: 20002

   Segment 2 → sequence number: 21002

   Segment 3 → sequence number: 22002

   Segment 4 → sequence number: 23002

   Segment 5 → sequence number: 24002

4) **Acknowledgement Number**: This is used by the sender to acknowledge the received data. This 36-bit field indicates the sequence number of the next byte expected from the receiver. For example, if host A has sent a segment having sequence number 2000 to host B, host B would send an acknowledgement with acknowledgement number field set to 2001 (one plus the sequence number of last received segment).

5) **Header Length**: The HLEN field consists of 4 bits. It indicates the length of the TCP header. The length of the TCP header can be between 20 bytes to 60 bytes i.e., HLEN field can have binary values ranging from 0101-1111 (5 to 15, 32 bit words) (5 x 4 =20, 15 x 4 =60).

6) **Reserved**: This 6 bit field is reserved for future use. The value set in this field must be zero.

7) **Control Flags***:* This field contains six different control flags that can control certain aspects of the TCP connection such as connection establishment, connection termination and flow control. The flags include:

   a) **Urgent Pointer URG**: When set, the ACK indicates that the current segment contains urgent (or high-priority) data and that the Urgent Pointer field value is valid.

   b) **Acknowledgement (ACK)**: When set, indicates that the value contained in the Acknowledgement Number field is valid. This bit is usually set, except during the first message during connection establishment.

   c) **Push (PSH)**: PSH is used when the transmitting application wants to force TCP to immediately transmit the data that is currently buffered to the application without waiting for the buffer to fill. It is useful for transmitting small units of data.

   d) **Reset (RST)**: When set, RST immediately terminates the end-to-end TCP connection.

   e) **Synchronize (SYN)**: SYN is set in the initial segments used to establish a connection, indicating that the segments carry the initial sequence number.

   f) **Finish (FIN)**: FIN is set to request normal termination of the TCP connection in the direction this segment is travelling. Complete closure of the connection requires one FIN segment in each direction.

8) **Window Size**: The window size 16 bits field is used for flow control. It contains in bytes, the size of the window that the receiver has to maintain i.e., the value of the *receive window size.* It is basically the number of transmitted bytes that the sender of this segment is willing to accept from the receiver.

9) **Checksum**: This is a 16-bit field that provides bit error detection for the segment (including the header and data).

10) **Urgent Pointer**: Urgent data is information that has been marked as high-priority by a higher layer application. The data sent under high-priority usually bypasses the normal TCP buffering and is placed in a segment between the header and normal data. When the URG flag of control flag is set, then the urgent pointer 16-bit number indicates the position of the first octet of non-priority data in the segment.

11) **Options:** The option field contains 40 bytes of optional information about connection establishment. The maximum segment size (MSS) is the most commonly used option and if absent, defaults to an MSS of 536. Another option is Selective Acknowledgement (SACK), which allows out-of-sequence segments to be accepted by a receiver. The further discussion about options is beyond the scope of this book.

Characteristics of TCP

The basic characteristics of TCP are as follows:

1) It employs a connection-oriented service for communication.
2) It is a reliable source of communication i.e. guarantees delivery of messages.
3) It splits the messages into segments and keeps track of the order (sequence) of segments.
4) It employs the checksums for detecting any errors in data as well as the TCP header.

3.3.2 TCP Connection Establishment and Termination

A connection is a requirement of a reliable data delivery service. It is set up before the actual data exchange takes place. The connection is used to acknowledge the receipt of packets and retransmit those that are lost.

TCP provides a connection-oriented service over packet switched networks. 'Connection-oriented' implies that there is a virtual connection between two endpoints. There are three phases in any virtual connection. These are the connection establishment, data transfer and connection termination phases.

3.3.3 TCP Connection Establishment

In order to connect two machines, first of all both machines must initialize the communication and get a formal approval from each other before the start of data transfer. The establishment of such a connection by exchange of control messages is known as the three-way handshake. The process of the three-way handshake is as under:

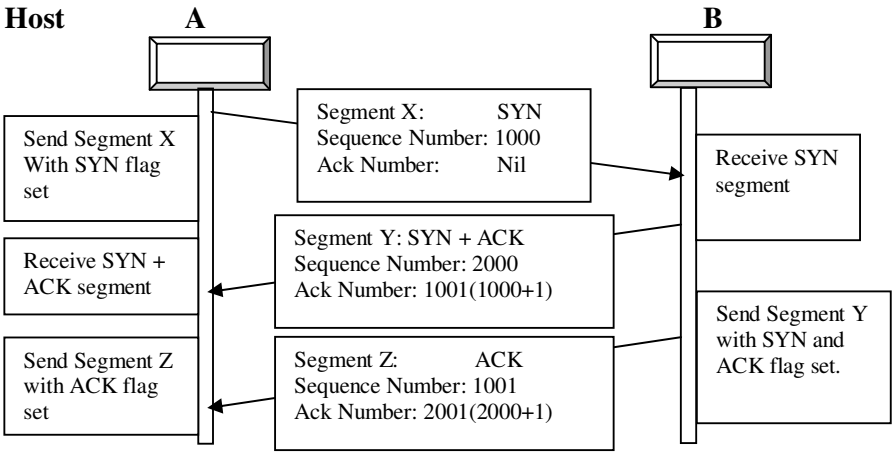


Figure 5: TCP Connection Establishment

1) The initiator of the session (source machine) sends a segment with SYN flag set to the destination node along with the proposed initial sequence number (ISN) in the sequence number field of the segment. Remember, the segment includes the source and destination port numbers. For example, in *Figure 5*, Host A sends a segment named X, with SYN flag set, initial sequence number is 1000 and acknowledgement number is Nil.

2) Upon receipt of the segment, the recipient sends a segment with SYN flag set to the initiator with the acknowledgement number set to the sequence number (ISN) + 1 and the ACK flag is also set. The sent segment is assigned a new sequence number (ISN) for its own segment. For example, in *Figure 5*, Host B, after receiving the segment, sends a segment named Y with SYN + ACK flag set, initial sequence number assigned to the segment is 2000 and acknowledgement number is 1001(sequence number of segment sent by host A plus 1).

3) The initiator after receiving the segment sends a segment with ACK flag set in response to the recipient's SYN, with the acknowledgement number set to the recipient's sequence number (ISN) + 1. For example, in *Figure 5*, Host A, after receiving the segment, sends a segment named Z, with ACK flag set, sequence number assigned to the segment is 1001(the first segment sent by A had a sequence number of 1000) and acknowledgement number is 2001(sequence number of segment sent by host B plus 1).

3.3.4 TCP Connection Termination

There is a saying that, "all good things must come to an end" and so it is with TCP connections. After the exchange of data between the source and destination nodes, either of them can close the connection. Thus, after the user has sent all its data and wishes to close the TCP connection, four segments are required to completely close a connection gracefully from both directions. The connection termination phases are discussed below:

1) The initiator (source) sends a segment with the FIN flag set. For example, in *Figure 6*, Host A sends a segment named X, with FIN flag set, initial sequence number is 2000 and acknowledgement number is Nil.

2) Upon receipt (destination) of the segment, the recipient issues an ACK segment, in order to confirm the receipt of the initiator. The acknowledgement number is 1 plus the sequence number received from the initiator i.e. from the FIN segment. For example, in *Figure 6*, Host B, after receiving the segment, sends a segment named Y, ACK flag set, initial sequence number assigned to the segment is 4000 and ack no. is 2001(sequence number of segment sent by host A plus 1).

3) If the recipient (destination) still has some data to send to the initiator (source), it sends the data. Otherwise, the recipient (destination) sends a request for termination of connection from its side by sending a segment with FIN flag set.

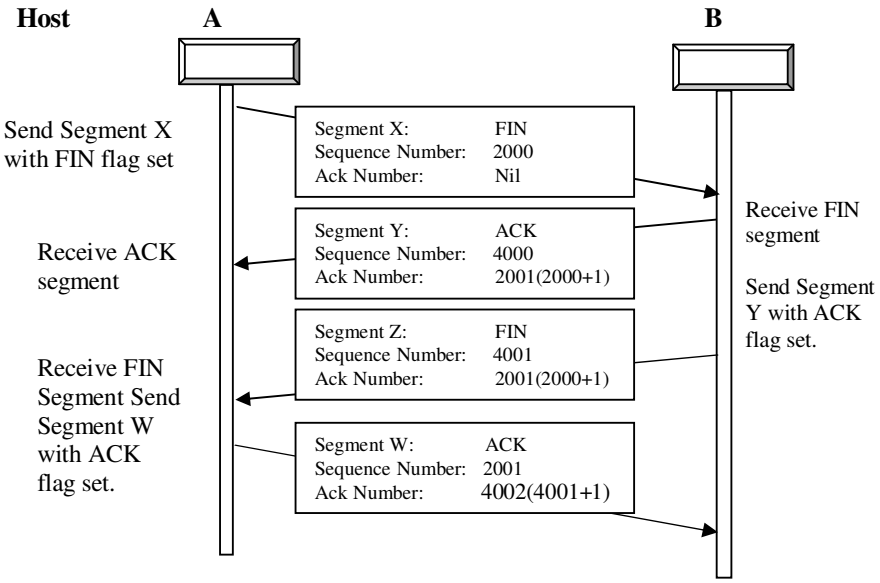


Figure 6: TCP Connection Termination

For example, in *Figure 6*, in case Host B too wishes to disconnect, it sends another segment named Z, with FIN flag set, sequence number assigned to the segment is 4001(sequence number of the last segment sent by B plus 1) and acknowledgement number is 2001(sequence number of segment sent by host A plus 1).

4) On receipt of this segment, the initiator sends an ACK segment, to confirm the FIN segment from the recipient (destination). For example, in *Figure 6*, Host A, after receiving the segment, sends a segment named W, with ACK flag set, sequence number assigned to the segment is 2001(the first segment sent by A had a sequence number of 1000) and acknowledgement number is 4002(the last sequence number of segment sent by host B plus 1).

Another method of closing the connection can be done by requesting the resetting of connection i.e., a host machine can send a segment with the RST bit set which will convey the other machine to immediately terminate the connection. Apart from

connection establishment and connection termination, flow control, error control and congestion control are few other responsibilities being performed by TCP.

Flow control in TCP

The flow control coordinates the amount of data that can be sent by the source between the terminals before receiving acknowledgement from the destination.

In order to speed up the communication, the source can send the complete message at a given instance without waiting for acknowledgement to arrive from the receiver. However, such a situation can cause adverse effects at the receiver side. In such situations, TCP sends data in accordance with sliding window protocol – a Flow Control Mechanism.

Sliding Window Protocol

In the sliding window protocol, a window is maintained for each connection. The window defines the size of the buffer i.e. the total number of bytes that can be sent by a terminal at a given time shown in *Figure 7,* this also shows the total number of blocks signifies the total size of the window. The sliding window also keeps track of bytes which have been sent but are unacknowledged, bytes still stored in buffer but haven't been sent etc.

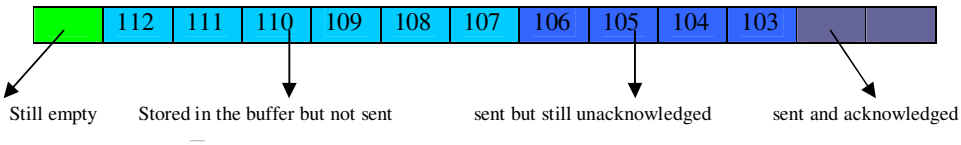


Figure 7: Sliding Window of sender node

Initially a window size is negotiated between the end terminals especially from the receiving side, while establishing a connection. Since TCP provides a byte-stream connection, sequence numbers are assigned to each byte in the stream. TCP divides this contiguous byte stream into TCP segments to transmit them. Therefore, the window principle is used at the byte level, that is, the segments sent and acknowledgements received will carry byte-sequence numbers and the window size is expressed as a number of bytes.

There are mainly two windows that are maintained i.e., one by the sender and another one by the receiver. The receiver window stores information about the various bytes received by its buffer but are still unconsumed and the other empty locations of the buffer.

With the help of these windows, a maximum limit on the number of bytes that can be sent by the sender to the receiver is decided, depending upon the total number of bytes that can be stored in the receiver's buffer at that particular instance. Thus, the mechanism of sliding window helps in controlling the flow of packets.

Error Control in TCP

The major tasks of error control in TCP include detection of out-of-order segments, lost segments, duplicate segments and corrupted segments. The above-mentioned problems are deciphered using the aid of acknowledgements, checksum and time-out period.

Out-of-Order segment

As IP is a connectionless service, IP datagrams might arrive out of order. The TCP in turn assures that the receiver does not acknowledge the received out of order segment until and unless it receives all those expected segments which precede it as shown in *Figure 8.*

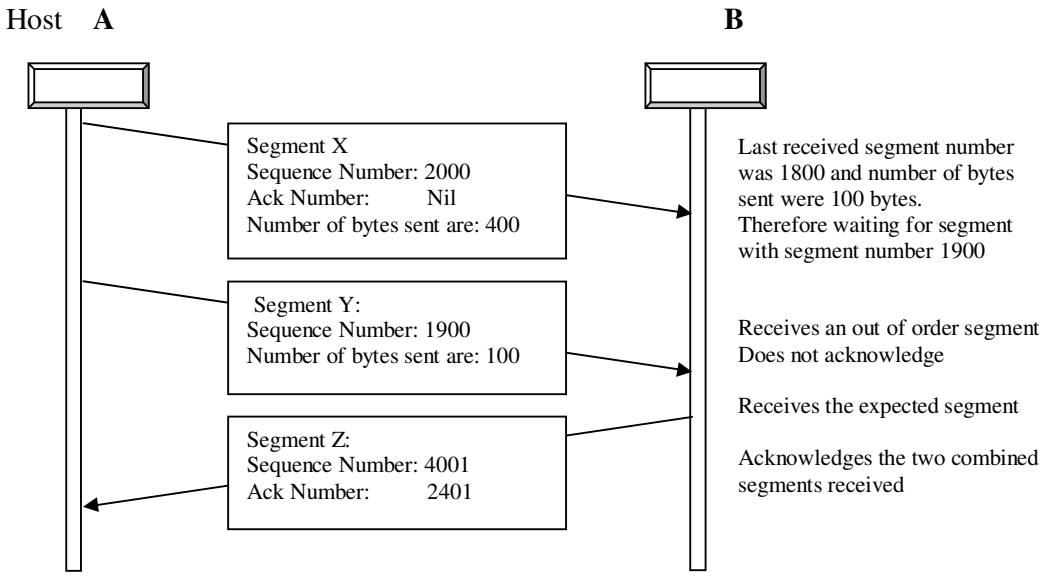

Host A
B
Segment X
Sequence Number: 2000
Ack Number: Nil
Number of bytes sent are: 400
Last received segment number was 1800 and number of bytes sent were 100 bytes.
Therefore waiting for segment with segment number 1900
Segment Y:
Sequence Number: 1900
Number of bytes sent are: 100
Receives an out of order segment
Does not acknowledge
Receives the expected segment
Segment Z:
Sequence Number: 4001
Ack Number: 2401
Acknowledges the two combined segments received

Figure 8: Out of order segments

Duplicate Segment

If the receiver receives a duplicate segment, it simply discards that segment, as a TCP segment with the same sequence number has already arrived.

Corrupted Segments

Whenever a segment gets corrupted, it is simply discarded by the destination and has to be retransmitted by the source as shown in *Figure 9*.

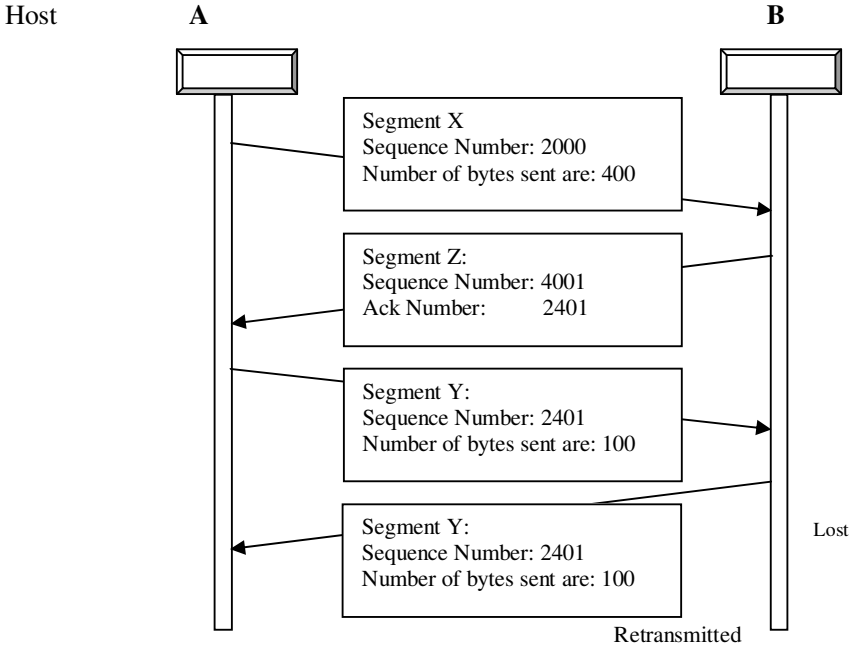


Figure 9: Corrupted Segment

Lost Acknowledgements

An acknowledgement for a segment can be lost in between. However, in TCP, the functioning of acknowledgement is as follows: An acknowledgement is a confirmation that everything up to the bytes specified by the acknowledgement

number in TCP header has been received. Thus, a latest acknowledgement overrides the previous reached /lost acknowledgements as shown in *Figure 10*.

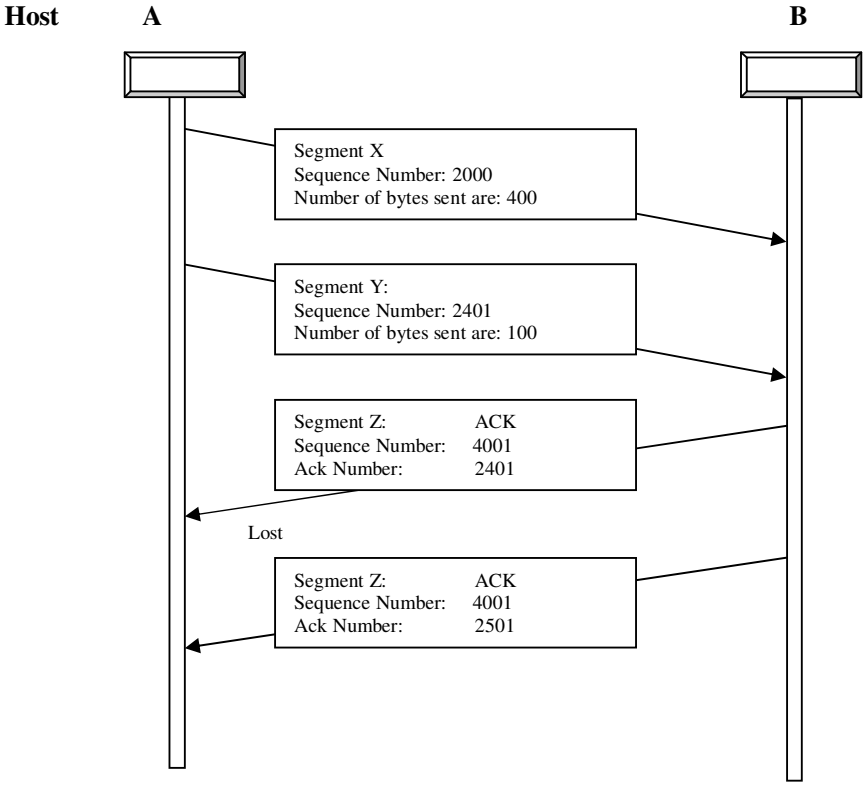


Figure 10: Lost Acknowledgement

Check Your Progress 1

1) Which of the following are transport layer protocols?

   a) TCP

   b) UDP

   c) IP

   d) Both a and b

   ……...

2) The combination of IP address and port numbers is known as-

   a) Socket Address

   b) Passive Address

   c) Transport Address

   d) Network Address

   ……...

3) Which of the following function is performed by the TCP.

   a) Process-to process communication

   b) End to end delivery of data

   c) Host to host communication

   d) None of the above

   ……..

4) How many types of port numbers are used?

   a) One

   b) Two

   c) Three

   d) None of the above

   ……...

5) Which of the following are parts of TCP header?

   a) Sequence Number

   b) Acknowledgement Number

   c) Checksum

   d) All of the above.

   ……...

3.4 USER DATAGRAM PROTOCOL (UDP)

UDP is an unreliable transport layer based protocol. It uses the connectionless service for providing the communication between the nodes of similar and/or different networks. The IP layer provides host-to-host communication. However, UDP provides process-to-process communication.

As UDP is connectionless, it means that no overheads will be incurred in terms of connection establishment, connection termination, acknowledgement of packets, sequence number assigned to each packet etc. Similar to the services offered by IP, UDP does not provide flow control and error control. In UPD, whenever the message is ready for transmission, it is simply transferred to the lower layers without bothering about establishment of connection. Thus, in situations where a small message has to be transmitted, UDP should be employed. On the other hand, the UDP is an unreliable source of communication. It does not guarantee accuracy, reachability and timeliness. Sequence numbers are not assigned to the packets, therefore, order of packets is not maintained and packets are unacknowledged.

UDP is a convenient protocol for multimedia and multicasting applications. It is suitable for processes that require simple request – reply communication with little interest in flow and error control.

UDP Header

The UDP packets are termed as *User Datagrams* and the header part of the user datagram has a fixed size of 8 bytes as shown in *Figure 11*.

1) **Source Port Number:** A 16-bit number which defines the source port number for a particular application program that is sending the UDP datagrams.

2) **Destination Port Number:** A 16-bit number that defines the destination port number for a particular application program that is receiving the UDP datagrams.

Source port number 16 bits	**Destination port number 16 bits**
Length 16 bits	Checksum 16 bits
Data	

Figure11: UDP Header

3) **Length**: The 16-bit field denotes the size of UDP header combined with payload data. It can range between 0 to 65,535 bytes.

4) **Checksum**: This is 16-bit field that provides detection of errors over the entire user datagram.

Characteristics of UDP

The basic characteristics of UDP are as follows:

1. UDP is a connectionless service.
2. It adds no non-reliable flow control to IP.
3. It serves as a multiplex/demultiplexer for sending and receiving datagrams.
4. The communication ends (end terminals) need not be synchronized.
5. There is no provision for acknowledgement of datagrams.

Applications of UDP

The standard applications using UDP include:

- Trivial File Transfer Protocol (TFTP)
- Domain Name System (DNS) name server
- Remote Procedure Call (RPC) used by the Network File System (NFS)
- Simple Network Management Protocol (SNMP).

Internet Control Message Protocol (ICMP)

The Internet Control Message Protocol notifies the sender of IP datagrams about abnormal events. ICMP is important in the connectionless environment. ICMP messages are carried in IP packets. The most commonly employed ICMP message types include:

- **Destination Unreachable**: This message type indicates that a packet cannot be delivered because the destination host cannot be reached. The reason for the non-delivery may be that the host or network is unreachable or unknown, the protocol or port is unknown or unusable.
- **Echo and Echo Reply**: These two messages are used to check whether hosts are reachable on the network. One host sends an Echo message to the other, optionally containing some data and the receiving host responds with an Echo Reply containing the same data. These messages are the basis for the Ping command.
- **Source Quench**: Sent by a router to indicate that it is experiencing congestion and is discarding datagrams.
- **TTL Exceeded**: This message indicates that a datagram has been discarded because the TTL field reached 0 or because the entire packet was not received before the fragmentation timer expired.
- **Timestamp and Timestamp Reply**: These messages are similar to the Echo messages, but place a timestamp (with millisecond granularity) in the message, yielding a measure of how long remote systems spend buffering and processing datagrams and providing a mechanism so that hosts can synchronize their clocks.

☞ Check Your Progress 2

1) The UDP performs which of the following function:

   a) Process-to process communication

b) End to end delivery of data

c) Host to host communication

d) None of the above

……

……

2) Which of the following are parts of UDP header?

a) Sequence Number

b) Acknowledgement Number

c) Checksum

d) All of the above

……...

3) Which of the following field (s) is/are required for ordering the packets?

a) Sequence Number

b) Acknowledgement Number

c) Checksum

d) All of the above

……...

4) What is the purpose of sliding window protocol?

a) Congestion Control

b) Flow Control

c) Error Control

d) All of the above

……...

5) How is TCP different from UDP?

……...

3.5 SUMMARY

This unit provides a complete overview of TCP. We have learnt that the data link layer provides node-to-node (intermediate), network layer provides host-to-host and transport layer provides process-to-process communication. The transport layer uses TCP and UDP protocols. TCP is a reliable, connection-oriented service, while UDP is an unreliable and connectionless service. As real communication takes place between two application programs, each process should be assigned a unique port number for identifying the various processes running on the same machine. There are three types of port numbers: well known, registered and dynamic ports. The combination of IP address and port number is called a socket address. The size of UDP header is 8 bytes and TCP header requires 20-60 bytes. The TCP employs error control, flow control and congestion control mechanisms. A three-way handshake is required for TCP connection establishment. The TCP connection termination requires four steps. The next unit of this course covers the application layer protocols and their role in TCP/IP.

3.6 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) D
2) A
3) A
4) C
5) D

Check Your Progress 2

1) A
2) C
3) A
4) B
5) TCP is a connection-oriented service and UDP is a connectionless service.

3.7 FURTHER READINGS

1) Andrew S. Tanenbaum, *Computer Networks*, Third edition.

2) Behrouz A. Forouzan, *Data Communications and Networking*, Third edition.

3) Douglas E. Comer, *Internetworking with TCP/IP Vol.1*: Principles, Protocols, and Architecture (4th Edition).

4) James F. Kurose, *Computer Networking*: *A Top-Down Approach Featuring the Internet* (3rd Edition).

5) Larry L. Peterson, *Computer Networks*: *A Systems Approach*, 3rd Edition (The Morgan Kaufmann Series in Networking).

6) W. Richard Stevens, *The Protocols (TCP/IP Illustrated, Volume 1).*

7) William Stallings, *Data and Computer Communications*, Seventh Edition.

UNIT 4 APPLICATION LAYER PROTCOLS

Structure **Page Nos.**

4.0 Introduction 76

4.1 Objectives 76

4.2 Domain Name System (DNS) 77

4.2.1 Hierarchical Name Space 77

4.2.2 Domain Servers 78

4.2.3 How does DNS Work in Internet 79

4.2.4 Domain Name Resolution 80

4.2.5 Messages Used in DNS 80

4.2.6 Dynamic DNS (DDNS) 82

4.3 Electronic Mail 83

4.3.1 Simple Mail Transfer Protocol (SMTP) 83

4.3.2 Message Transfer Agent 83

4.3.3 User Agent 84

4.3.4 Post Office Protocol (POP) 85

4.3.5 Internet Mail Access Protocol (IMAP) 85

4.3.6 Multipurpose Internet Mail Extension (MIME) 85

4.4 Telnet 87

4.5 File Transfer Protocol (FTP) 88

4.6 Summary 92

4.7 Answers to Check Your Progress 92

4.8 Further Readings 93

4.0 INTRODUCTION

The application layer provides means for the user to access information on the network through an application. This layer is the main interface/environment for the user to interact with the application and therefore with the network. The application layer can be assumed to be the level of TCP/IP protocol hierarchy where the user-accessed network processes reside. The applications running at the application layer are some network processes that occur above the transport layer. This includes all the processes that the users can interact with. Thus, application layer provides the services user applications communicate through the network for e.g., SMTP, FTP, Telnet and Rlogin. In this unit, we shall discuss the various standard services such as DNS, FTP, Telnet and SMTP being offered by the application layer.

4.1 OBJECTIVES

Our objective is to introduce you to the basic concept of the application layer and its protocols. On successful completion of this unit, you should be able to:

- have a reasonable understanding of the application layer;
- describe the operation of application layer protocols;
- understand the role and meaning of Domain Name System (DNS); and
- describe and understand the working of Simple Mail Transfer Protocol (SMTP), Telnet and File Transfer Protocol (FTP).

4.2 DOMAIN NAME SYSTEM (DNS)

As we already know, each host machine on the Internet is assigned an IP address. The 32-bit IP address is represented in the numerical form, e.g., 202.12.32.22. However, it is very difficult for a user to remember the address of a machine in terms of an IP address. Therefore, the IP addresses have been denoted as a set of characters in English e.g., for the other name, of IP address 68.142.226.32 is yahoo.com. This human readable name in place of an IP address is known as domain name, e.g., google.com. The domain names are registered by an organization called ICANN. The domain names are not case sensitive and can contain alphabetic or numeric letters or the hyphen. Whenever a user accesses the Internet, he or she types the domain name instead of an IP address. Therefore, there is a need for a system that can convert the domain names of hosts into IP addresses and *vice-versa*. Such conversions are carried out by a system known as Domain Name System. The main function of Domain Name System (DNS) is to map the IP addresses into domain names (human readable names).

4.2.1 Hierarchical Name Space

The DNS is a large database that resides on various computers and it contains the IP addresses of various hosts on the Internet and various domains. The Domain Name System (DNS) is basically a distributed Internet directory service. In this system, the host needs to map the closest DNS computer machine which has the required information.

In order to assign the names to various machines, some kind of naming system was required. Thus, a hierarchical name space was designed such that each domain name was divided into several parts of the tree, e.g., the domain name 'yahoo.com' consists of two parts. First the domain name is read from right to left, i.e., com signifies a commercial website (upper level of hierarchical system) and yahoo denotes the name of the website (lower level of hierarchical system) see *Figure 1.*

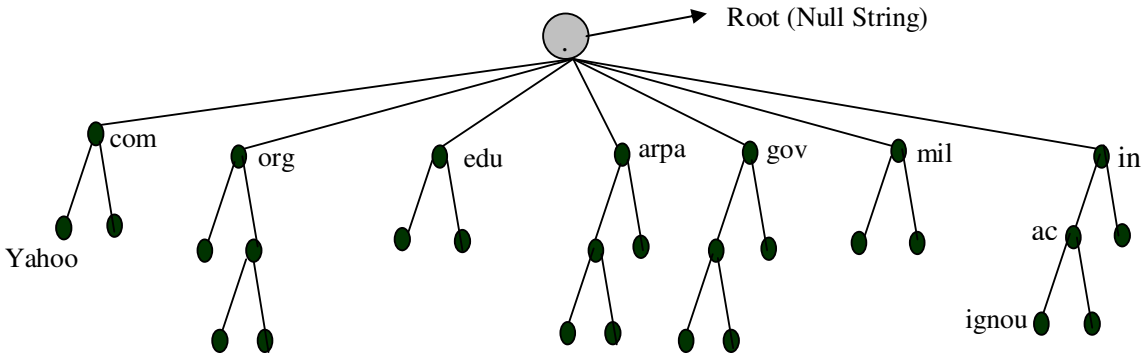


Figure 1: Hierarchical Name Space for DNS

In DNS, a tree structure has been designed such that root of the tree binds the complete tree. The maximum levels of the tree are 128 and each label of the tree can have a string of 63 characters. The root of the DNS hierarchy (tree) is designated with a period '.'. Thereafter the tree contains a group of top-level domains including familiar names like *com, org, edu*, various country-level domains like *in* (India) etc. However, the domain names at this level cannot be assigned to an individual machine. The next levels of the DNS tree are the second-level registered domains such as hotmail.com. Below the second level, local domains names can be specified. Remember, the domain names are always read from the current node up to the root of the tree as shown in *Figure 2*.

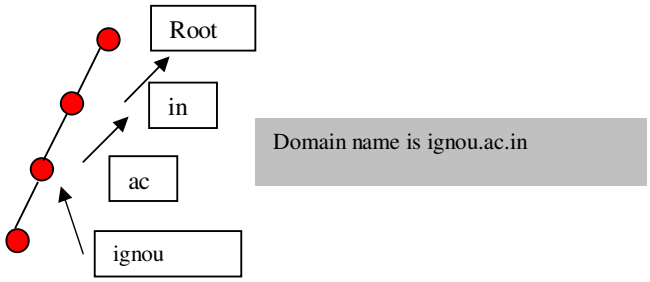


Figure 2: Labeling for domain names

The above mentioned tree structure has to be stored into the computer machine. However, storage of such a huge amount of information on a single computer system is prone to inefficient computing due to the following reasons:

1. Availability of the system.
2. Reliability.
3. Computing capacity of the system.
4. Network load on that particular transmission link.

4.2.2 Domain Servers

Due to these reasons, domain name information has been distributed among various servers known as Domain Servers. The server is responsible or has an authority over a specific region called a zone i.e., the DNS database is divided into zones. The servers in their respective zones are responsible for answering queries for their zones and are called name servers.

A name server is a server program that holds a master or a copy of a name-to-address mapping database, or otherwise points to a server that does and that answers requests from the client software, called a name resolver. Conceptually, all Internet domain servers are arranged in a tree structure that corresponds to the naming hierarchy.

A zone is simply a sub-tree of DNS and is administered separately. There are multiple name servers for a zone. There is usually one primary name server and one or more secondary name servers. A name server may be authoritative for more than one zone.

The root server is the server whose zone consists of the complete tree. Basically a root server does not keep information about domain names but actually delegates its own authority to other servers. At present, there are around 13 root servers distributed all over the world which are able to cover the entire set of domain names.

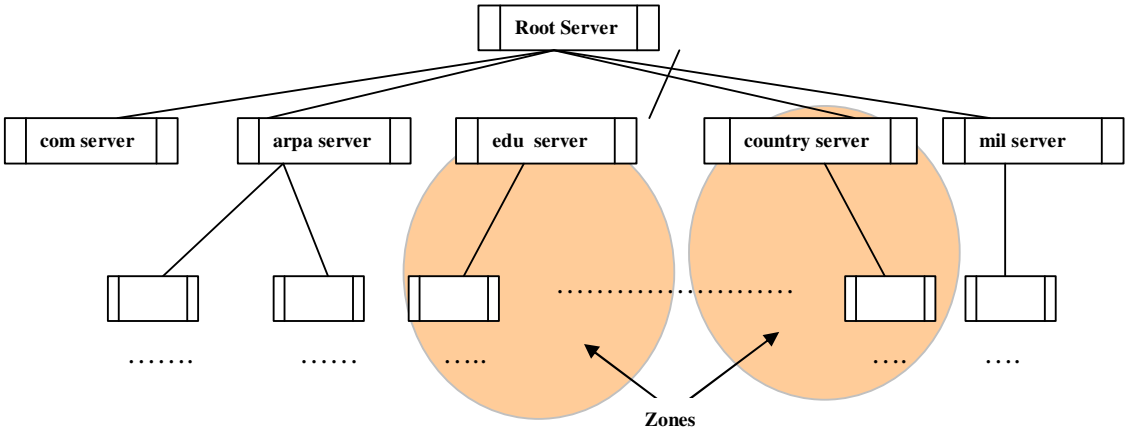


Figure 3: Name Servers

Thus, we have discussed in general the working of Domain Name Systems.

4.2.3 How does DNS Work in Internet?

The *hierarchical* system of domain names in Internet has been broadly classified into three categories:

1. Generic domain names
2. Country based domain names
3. Inverse domains

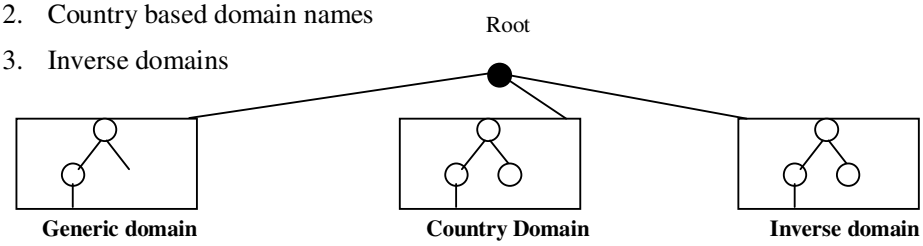


Figure 4: Hierarchy of Domains in Internet

The three-character top-level names are called the generic domains or the organisational domains. The generic domain defines registered hosts in accordance with their behaviour for example, yahoo.com uses 'com' as top level name as it signifies that yahoo is a commercial organisation. *Table 1* shows some of the top-level domains of today's Internet domain hierarchy.

Table 1: **Generic Domain Names**

Domain Name	Meaning
Com	Commercial organisations
Gov	Government institutions
Org	Non-profit organisations
Mil	Military groups
Edu	Educational institutions
Net	Major network support centers
Int	International organisations

There are also top-level domains named for each of the ISO 3166 international 2-character country codes (e.g., zw for Zimbabwe). These are called the country domains. Many countries have their own second-level domains underneath, which parallel the generic top-level domains.

As it is a simple matter in principle to search the database for an IP address with its symbolic name (because of the hierarchical structure), the reverse process cannot follow the hierarchy. Therefore, there is another namespace for the reverse mapping called Inverse domain. The information about the inverse domain is always stored in the domain "in-addr.arpa". Here, the IP addresses are stored in a hierarchical form such that each level of the tree depicts 8 bits of information of IP address. However, since domain names have the least significant parts of the name first, but dotted decimal format has the most significant bytes first, the dotted decimal address is shown in reverse order. For example, the domain in the domain name system corresponding to the IP address 171.10.1.2 is 2.1.10.171.in-addr.arpa. The above mapping is carried out by using a special kind of query called inverse pointer query.

4.2.4 Domain Name Resolution

The concept of mapping a domain name to an IP address and *vice-versa* is known as resolution process. The resolution process is basically a client server platform. Whenever a user needs to map an address to a domain or *vice-versa*, the DNS calls a client program called resolver. The resolver subsequently contacts the nearest DNS server (name server) with a request. In case the server has the desired information, it replies back with the results. Otherwise it suggests the resolver to other domain servers or asks other servers to provide the desired information. The resolver, after receiving the results asserts the information and thereafter delivers the desired information to the specific host process. The steps followed in the resolution are below:

1) The user program issues a request such as the gethostbyname( ) system call. *(This particular call is used to ask for the IP address of a host by passing the host name as a parameter).*
2) The resolver formulates a query to the name server.
3) The name server checks to see if the answer is in its local authoritative database or cache, and if so, returns it to the client. Otherwise, it will query other available name server(s), starting down from the root of the DNS tree or as high up the tree as possible.
4) The user program will finally be given a corresponding IP address (or host name, depending on the query) or an error if the query could not be answered.

As the resolution process is carried out with the help of queries, these domain name request queries can be one of two types: *recursive* or *iterative*. A flag bit in the domain name query specifies whether the client desires a recursive query, and a flag bit in the response specifies whether the server supports recursive queries. A recursive query requests that the server should itself issue a query to determine the requested information and return the complete answer to the client. However, an iterative query means that the name server should return what information it has available and also a list of additional servers for the client to contact to complete the query.

The concept of caching is also utilised in DNS. Whenever a name-server receives a request from a client, the name server subsequently sends the query to other servers. Later on, when it receives the response, it first stores this information in its own cache memory before sending the desired information to the client. Now onwards, if any client desires the same information, first the name-server can simply check its cache memory and resolve the query. However, such a response is designated as non-authoritative. The domain name responses from the name server can be one of two types: authoritative and non-authoritative.

4.2.5 Messages used in DNS

As DNS follows the client server paradigm, it has two types of messages: Query and Response. The messages in the DNS use a single format i.e., similar format. The general format of a query message and that of a response message are shown in *Figures 5* and *6*.

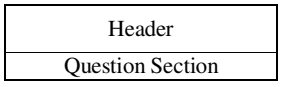


Figure 5: Query Message

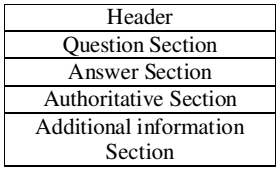
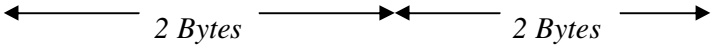


Figure 6: Response Message

The format of the header of the message is shown in *Figure 7*. The header section is always present and has a fixed length of 12 bytes.



Identification	Flags
Number of Question Records	Number of Answer Records
Number of Authoritative Records	Number of Additional Records

Figure 7: Header format for Query/Response

- **Identification** used by the source of the message in order to match the response of the query.
- **Flags** contains various fields that define the kind of message i.e., recursive, iterative, authoritative, non-authoritative etc.
- **Number of Question Records** contains the total number of queries asked by the resolver in the Question Section.
- **Number of Answer Records** contains the total number of answers specified in the Answer Section.
- **Number of Authoritative Records** contains the total number of Authoritative records in the Authoritative Section.
- **Number of Additional Records** contains the total number of Additional records in the Additional Section.

It may be noted that the Number of Answer Records, Number of Authoritative Records, Number of Additional Records sections will have ZERO value in the query message.

Example

Now let us take up an example and illustrate the complete procedure of mapping a domain name to an IP address.

1) Let us say a user opens a web browser and tries to connect to a website, say www.ignou.ac.in.
2) The operating system not knowing the IP Address for www.ignou.ac.in, asks the (Internet Service Provider) ISP's DNS Server for this information.
3) The ISP's DNS Server does not know this information, so it connects to a Root Server to find out what name server, running somewhere in the world, knows the information about ignou.ac.in.
4) The Root Server tells the ISP's DNS Server to contact a particular name server that knows the information about ignou.ac.in.
5) Thereafter, the ISP's DNS Server connects to IGNOU's DNS server and asks for the IP Address for www.ignou.ac.in.
6) IGNOU's DNS Server responds to the ISP's DNS server with the appropriate IP Address.
7) The ISP's DNS Server tells the User's operating system the IP Address for ignou.ac.in.
8) The operating system tells the Web Browser the IP Address for www.ignou.ac.in.
9) The web browser connects and starts communication with www.ignou.ac.in.

4.2.6 Dynamic DNS (DDNS)

The Dynamic Domain Name System (DDNS) is a protocol that defines extensions to the DNS in order to enable the DNS servers to accept the requests to add, update and delete entries in the DNS database dynamically. Because DDNS offers a functional superset to existing DNS servers, a DDNS server can serve both static and dynamic domains at the same time. Rather than allowing any host to update its DNS records, the secure version of DDNS uses public key security and digital signatures to authenticate update requests from DDNS hosts.

Three common utilities for querying name servers are provided with many DNS implementations:

1) **Host** obtains an IP address associated with a host name or a host name associated with an IP address.
2) **Nslookup** allows you to locate information about network nodes, examine the contents of a name server database and establish the accessibility of name servers.
3) **Hig** allows you to exercise name servers, gather large volumes of domain name information, and execute simple domain name queries. DIG stands for Domain Internet Groper.

The following RFCs define the Domain Name System standard and the information kept in the system:

1) RFC 1032 – Domain Administrator's Guide
2) RFC 1033 – Domain Administrator Operations Guide
3) RFC 1034 – Domain Names – Concepts and Facilities
4) RFC 1035 – Domain Names – Implementation and Specification
5) RFC 1101 – DNS Encoding of Networks Names and Other Types.

☞ Check Your Progress 1

1) In the following domain names, identify the name using a country based domain name
   a) www.yahoo.com
   b) www.ignou.ac.in
   c) www.hotmail.com
   d) www.aol.com

   ……

   ……

2) The domain name resolution can be carried out through which of these methods?
   a) Iterative
   b) Recursive
   c) Caching
   d) All of the above

   ……

   ……

   ……

3) If the IP address of the machine is known and we want to enquire about its domain name, which one of the following is used?

   a) Generic domains

   b) Country domains

   c) Inverse domain

   d) None of the above

……

……

……

4.3 ELECTRONIC MAIL

Electronic mail (e-mail) is probably the most popular service of TCP/IP. For most people, it has become an integral part of everyday life. Electronic mail provides a platform for exchanging information between two end users. It is basically used for sending and receiving mails/messages, e.g., textual, voice, graphical and video messages between end users. This section provides an overview of the TCP/IP application protocol dealing with electronic mail.

4.3.1 Simple Mail Transfer Protocol (SMTP)

The standard mechanism for electronic mail in the Internet is *Simple Mail Transfer Protocol (SMTP)*. It provides mail and message exchange between TCP/IP hosts. SMTP is based on *end-to-end delivery* i.e., an SMTP client contacts the destination host's SMTP server directly for delivering the mail. The destination host's SMTP server keeps the mail until the mail has been successfully copied into the recipient's SMTP. The SMTP is a connection service based on client-server environment and runs on port number 25 at the server side as shown in the *Figure 8*. The various components of SMTP are:

1) Mail Transfer Agent (MTA)

2) User Agent (UA)

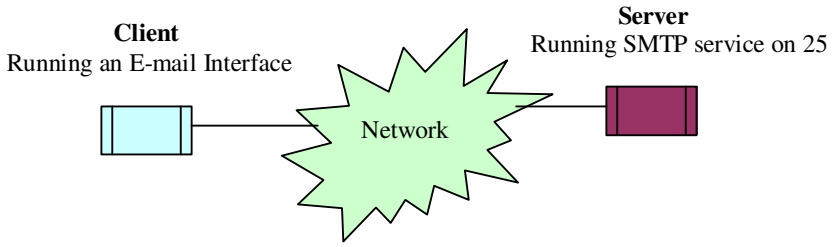


Figure 8: SMTP Service

4.3.2 Message Transfer Agent

The transfer of messages i.e., mails is delivered through an agent known as Message Transfer Agent (MTA). MTAs assist a user in sending as well as receiving the messages. The MTA consists of two shades i.e., MTA client for sending the mail while MTA server for listening/receiving the mails as shown in *Figure10*. The MTA is generic and defines how the commands and responses should be sent back and forth. A popular example of MTA in UNIX is known as sendmail.

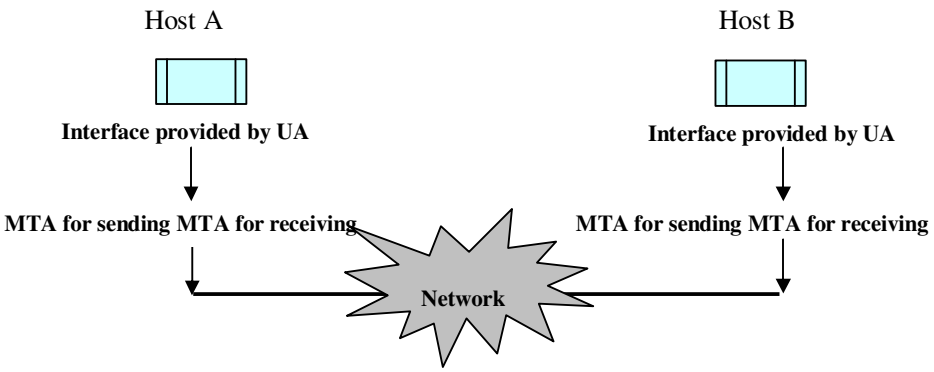


Figure 9: SMTP

4.3.3 User Agent

The user agent mainly deals with the composition of messages. The user agent provides an interface in the form as shown in *Figure 10* wherein s/he can write the message, specify the destination address (that is create an envelope). Therefore, UA puts the message into the envelope. The various services offered by the user agent are as under:

1) Reading the received messages
2) Replying to the read messages
3) Composing the messages
4) Forwarding the messages
5) Handling the various mailbox settings

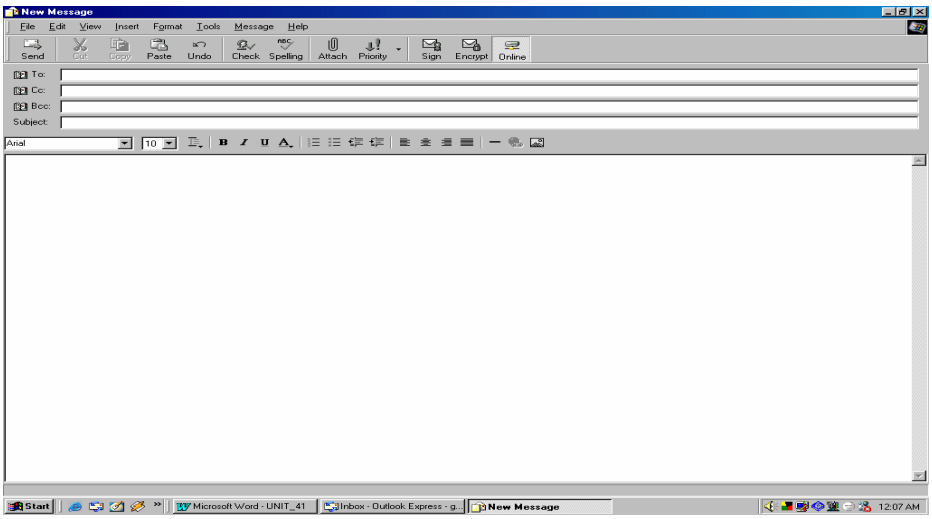


Figure 10: Interface provided by UA for composing a message

The address of a user (E-mail address) consists of two parts: the user name and the domain name. For example, amitgoel@yahoo.com is an e-mail address, wherein "amitgoel" depicts the user name, yahoo.com represents the domain name and these two parts are separated by an @ sign. Therefore, the mechanism of representing an e-mail address used by SMTP includes user name, @ and domain name. Remember that UA does not provide the facility for sending and receiving messages. The address of the destination host contains the domain name for identifying a particular network and thereafter a specific user on its network.

The interface presented by the user agents can be two types: command based or graphical interface based. The various examples of user agents with respect to command based interface are pine, mail etc. and with respect to GUI, some examples are: Netscape, outlook express etc.

4.3.4 Post Office Protocol (POP)

As SMTP is based on TCP/IP protocol, a TCP connection is required to be established between both ends. However, user machines cannot be expected to be online 24 x 7, especially a desktop machine owned by a home user. Therefore, there was a need for developing a system by which a user could receive his/her e-mails even though the machine was powered off. Therefore, most of the organisations install a SMTP server which is always online and receives e-mails on the behalf of each and every user of the organisation on its network. Basically the SMTP server acts as a "post office".

In order to retrieve the e-mails stored in the SMTP server on the behalf of the users, a protocol called *Post Office Protocol (POP)* has been devised. It assists in downloading the e-mails from the SMTP server as shown in *Figure 11*.

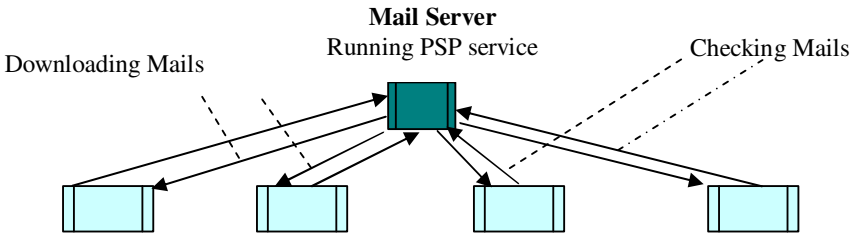


Figure 11: POP

4.3.5 Internet Mail Access Protocol (IMAP)

In POP, whenever a user accesses the mails from the mail server, i.e., downloads the mails, instantantly those accessed mails are deleted from the mail server. Thus, POP is not suitable for people accessing their mails from various locations, i.e., cyber cafe, home, hotel etc. POP does not provide the facility for creating folders, organising the mails on the mail server etc.

In order to avoid the deletion of the mails from the mail server, another protocol called Internet Mail Access Protocol (IMAP) has been devised. In addition to services offered by POP, IMAP provides the following services:

1) The user can create, rename or delete the mailbox on the mail server.

2) The user can check the header part of the mail before downloading the message

4.3.6 Multipurpose Internet Mail Extension (MIME)

The SMTP protocol sends the mails i.e., the messages only in a Network Virtual Terminal seven-bit ASCII format. That is, it will support only those languages that can be represented by a seven-bit ASCII format. Therefore, messages written in languages such as German, Russian and French cannot be sent through SMTP. Moreover, SMTP does not support the transfer of video files, audio files and binary files. Thus, there was a need for developing a mechanism for allowing the transfer of incompliant formats.

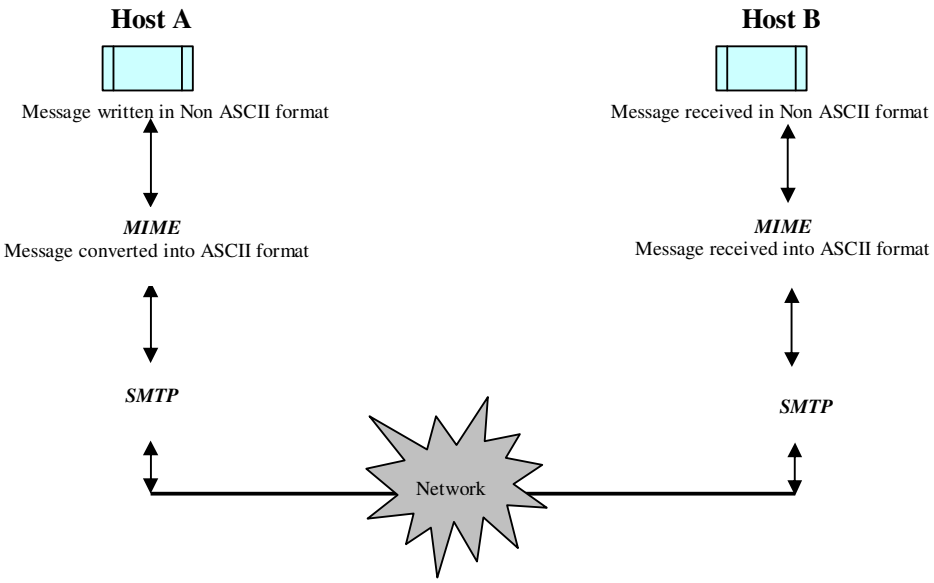


Figure 12: MIME

A protocol called Multipurpose Internet Mail Extension (MIME) supports transfer of Non-ASCII formats through SMTP. Primarily, MIME converts the non-ASCII formats into ASCII format and passes the data to SMTP. Consequently, the SMTP sends the ASCII form of data to the destination machine. The SMTP service at the destination machine passes the ASCII form of data to MIME which in turn converts the data into non-ASCII format as shown in *Figure 12*. Remember that MIME is simply an extension of SMTP and is not a mail protocol.

MIME defines five headers which can be supplemented with the original SMTP header in order to define the following parameters:

- **MIME Version:** It defines the version of MIME being used. The current version of MIME is 1.1
- **Content -Type:** It defines the type of data used in the body of the message, e.g., text, image, video, audio, postscript etc.
- **Content -Transfer-Encoding:** It defines the method to encode the mail message into 0's and 1's. The various kinds of encoding techniques are as follows: ASCII-7 bit, Non-ASCII-8 bit, Non-ASCII-Binary, Non-ASCII-Base64 and Non-ASCII-Quoted-printable.
- **Content-id:** It identifies the complete message in a multiple-message scenario.
- **Content-Description:** It describes whether the body of the message is text, image, video audio etc.

☞ Check Your Progress 2

1) What is the purpose of Message Transfer Agents?

   a) Creation of envelopes

   b) Transfer of messages

   c) Writing the messages

   d) None of the Above

   ……

   ……

2) Which field of MIME header describes the method to encode the message?

   a) Content-type
   b) Content-id
   c) Content-transfer-encoding
   d) Content-description

   ……

   ……

3) How is the Non-ASCII formats converted into ASCII format?

   a) MIME
   b) POP
   c) IMAP
   d) SMTP

   ……

   ……

4) Give a difference between POP and IMAP.

   ……

   ……

4.4 TELNET

The most fundamental method of data communication employed on a network is the ability to perform remote execution i.e. calling an application on a remote terminal. TELNET is a widely known application protocol that provides remote execution capability.

TELNET is a popular client server application program. TELNET is an abbreviation for ***TErminal NETwork.*** TELNET is a standard application protocol that provides an interface, through which a program on a host i.e., *TELNET client* can access the resources of another host i.e., *TELNET server*. TELNET provides an environment such that the client acts as a local terminal connected to the server as shown in *Figure 13*. Basically, TELNET is a utility whereby a user first logs into a remote machine and thereafter the user can access the files / programs located remotely.

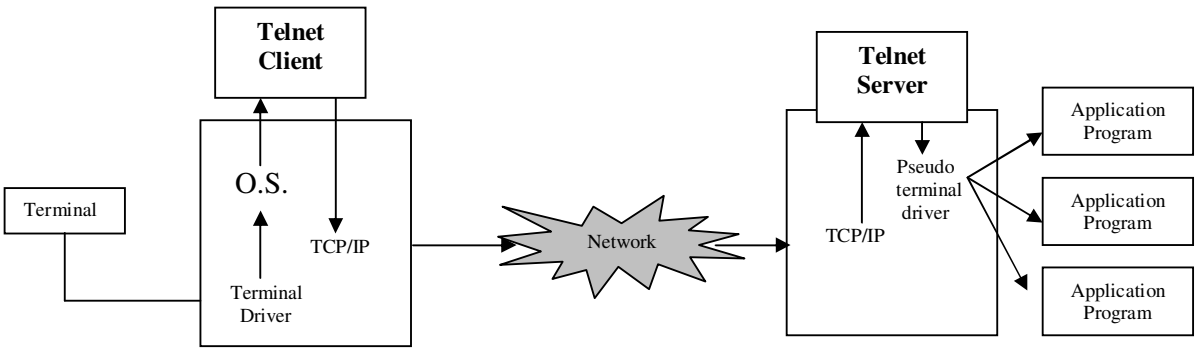


Figure 13: TELNET

The working of TELNET is as follows: The user types on the host machine /terminal that runs a driver known as terminal driver (a module of the operating system). It basically receives the keystrokes typed by the user on the terminal and passes the corresponding characters to the operating system. The operating system in turn sends these characters to the TELNET client. As discussed in SMTP, the language/format acceptable to the terminal might be different from the format allowed by TELNET. Therefore, the characters received by the TELNET client from the terminal driver are converted into a generic character set known as Network Virtual Terminal (NVT) characters. The transformed form of characters is sent to the TCP/IP protocol stack of the local machine.

The characters travel through the network using TCP/IP and finally reach the destination's operating system. The operating system passes the NVT characters to the TELNET server. The TELNET server in turn transforms the characters into characters understandable by the destination machine. The set of characters is sent to the operating system through a special kind of driver called pseudoterminal driver. As the TELNET server cannot directly interact with the operating system, the pseudoterminal driver helps the TELNET server in simulating a terminal. Thus, these characters from the operating system are delivered to the appropriate application program of the remote machine.

4.5 FILE TRANSFER PROTOCOL (FTP)

The transfer of files between two machines is one of the most common operations in a network. The standard mechanism for copying files from one machine to another is known as File Transfer Protocol (FTP).

Although it might seem quite simple to transfer data between two hosts, there are many issues which are to be resolved in such a transfer. For instance, two systems may use different file name conventions, different representation of data, different directory structures etc. These issues have been resolved with the help of FTP. FTP employs a client server environment. In order to access remote files in FTP, the user must firstly authenticate to the server before it allows the file transfer. Remember, FTP uses a connection-oriented service i.e., it is necessary to have both hosts up and running TCP/IP to establish a file transfer. After the establishment of connection, the data transfer between client and server takes place.

FTP uses TCP as a transport protocol to provide reliable end-to-end connection. The FTP server listens to connections on well-known port number 20 for transfer of data (data connection) and 21 for establishment / termination of connection (control connection). The control connection performs the task of authentication with the help of login and thereafter follows the TELNET protocol. As it is necessary to log into the remote host, the user must have a user name and a password to access files and directories.

The FTP application is built with a user interface at the top of the protocol, a Data Transfer Process (DTP) and protocol interpreter (PI) as shown in *Figure14*. On the client side, the user interface communicates with the protocol interpreter which is in charge of the control connection. This protocol interpreter (PI) has to communicate any necessary control commands to the remote system. On the Server side, the protocol interpreter, besides its function of responding to the TELNET protocol, has to initiate the data connection. During the file transfer, the data management is performed by DTPs. After a user's request is completed, the server's PI has to close the data connection.

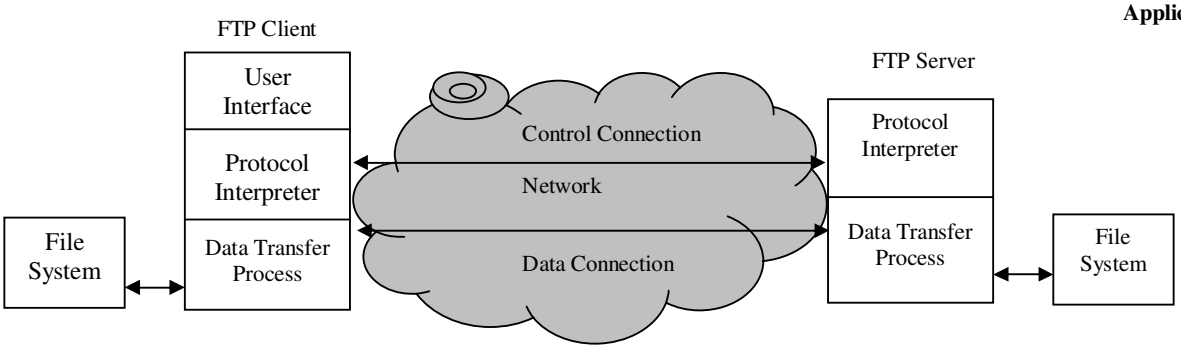


Figure 14: File Transfer Protocol (FTP)

The basic steps performed by a client during an FTP session are as given below:

1) **Connect the user to a remote host**

First the user has to authenticate the server machine about the user's authenticity with the help of a valid user name and password. The various commands that are required for connection establishment are:

- On the command prompt type "ftp".
- Then write a command "open" along with the IP address of the server machine, e.g., open 202.12.141.81. The open command imitates a login session on the remote machine.
- After connecting to the remote host, the remote machine enquires about the user name and its corresponding password.

  ftp> login

  ftp> password
- The other commands are user and pass. The purpose of user command is to identify a remote user id. The purpose of pass command is to authenticate the remote user id.

2) **Select a directory**

After connection is established, the client searches for the appropriate directory using "cd" (change directory) command. The user can select a local directory with the lcd (local change directory) command.

3) **List files available for transfer**

Commands like dir or ls are used for listing the various files in a particular folder depending upon the operating system.

4) **Define the transfer mode**

As the data transfer can take place between dissimilar systems, transformations of the data into an understandable form is required. The user has to mainly decide on two aspects of the data handling: first, the method through which the bits will be moved from one place to another and secondly, the different representations of data. In order to deal with the above issues, the following commands are used:

- **Mode**: It specifies whether the file is to be assumed as record structure in a byte stream format.
- **Type**: It specifies the character set being used for representation of data, i.e., ASCII, Image etc.

5) **Copy files to or from the remote host**

The following commands can be used to copy files between FTP clients and servers:

- **Get command:** It copies a file from the remote host to the local host.
- **Mget command:** It copies multiple files from the remote to the local host.
- **Put command:** It copies a file from the local host to the remote host.
- **Mput command:** It copies multiple files from the local host to the remote host.

6) **Disconnect from the remote host**

In order to disconnect a connection, the following commands can be used:

- **Close**: It disconnects from the remote host but leaves the FTP client running.
- **Quit / bye**: It disconnects from the remote host and terminates FTP.

FTP Illustration

Example 1

```
C:\] ftp www.ymcaie.ernet.in
Connected to ymcaie.ernet.in 220
ymcaie FTP server (Version 4.1) ready.
Name: Amit
Password required for Amit.
Password: ******
User Amit logged in.
ftp> put file.txt file1.txt
200 PORT command successful.
150 Opening data connection for file.txt (3453 bytes).
226 Transfer complete.
Local: file.txt remote: file1.txt 3453 bytes received in 0.082 seconds
ftp> close
221 Goodbye.
ftp> quit
```

Example 2

```
C:\] ftp www.ymcaie.ernet.in
Connected to ymcaie.ernet.in 220
ymcaie FTP server (Version 4.1) ready.
Name: Amit
Password required for Amit.
Password: ******
User Amit logged in.
ftp> ls /usr/amit
200 PORT command successful.
150 Opening ASCII mode.
226 Transfer complete.
ftp> close
221 Goodbye.
ftp> quit
```

Anonymous FTP

The first step required for establishing a connection with a remote host is a valid user id and password. However, there are few FTP servers that are available for general public access. Thus, in order to provide access to those public remote servers, a user need not have valid user id. *The user name for such servers is anonymous and password is guest.* Remember, the user access is limited in such cases.

Example 3

C:\] ftp www.ymcaie.ernet.in
Connected to ymcaie.ernet.in 220
ymcaie FTP server (Version 4.1) ready.
Name: anonymous
331 guest login OK, send "guest" as password
Password: guest
ftp> get file.txt file1.txt
200 PORT command successful.
150 Opening data connection for file.txt (3453 bytes).
226 Transfer complete.
Remote: file.txt Local: file1.txt 3453 bytes received in 0.072 seconds
150 Opening ASCII mode.
226 Transfer complete.
ftp> close
221 Goodbye.
ftp> quit

☞ Check Your Progress 3

1) Which command is used in ftp for connection establishment?

   a) open
   b) get
   c) put
   d) quit

   ……

2). Which command is used in ftp for transfer of data from local machine to remote machine?

   a) open
   b) get
   c) put
   d) quit

   ……

   ……

3). How many times is the control connection required in an FTP session?

   a) Once
   b) Twice
   c) Many times
   d) None of the above.

   ……

   ……

4) How many times is the data connection required in an FTP session?

   a) Once

   b) Twice

   c) As many times as required

   d) None of the above.

   ……

   ……

   ……

4.6 SUMMARY

This unit provides a complete overview of application layer of TCP/IP. Till now we have studied that there are various kinds of application layer related protocols. Some of them have been discussed in this unit, e.g., DNS, SMTP, TELNET and FTP. The domain name system is a client server based application that maps the domain names in the form of IP addresses. DNS maintains a hierarchical structure in order to store the information about a huge set of domain names. The name space is distributed among DNS servers. Every server is assigned a responsibility for a particular zone. There are 13 root servers located in different geographical regions. Each zone has its own primary server known as zone server. Whenever a user requests for an IP address, the name resolution, through a function called resolver, maps the domain name to the IP address and *vice-versa*.

The SMTP protocol provides a mechanism for providing a user interface and transfers the messages on the network. It has the following components: UA and MTA. The UA provides the interface for composing, reading, replying and forwarding the mail messages. The MTA performs the actual transfer of messages, on the network. MIME is a supplement to SMTP for allowing Non-ASCII format messages like video, audio etc. to be transferred through SMTP. POP and IMAP are extensions of SMTP that allow the messages to be stored into the mail server, so that later on, a user can access the mail server and read his/her mails.

TELNET protocol is a client server based application that provides remote login. FTP is a TCP/IP based application protocol employed for transferring files from one host to another. The basic commands required for connection establishment are: open, site etc. The commands used for data transfer are: *get, mget, put and mput*. The commands used for terminating a connection are: quit and close. The next block of this course covers the concepts of network programming.

4.7 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) B

2) D

3) C

Check Your Progress 2

1) B

2) C

3) A

4) POP downloads the mail messages on its client end and those mails are deleted automatically at the mail server end. However, IMAP avoids such deletions.

Check Your Progress 3

1) A

2) C

3) A

4) C

4.8 FURTHER READINGS

1) Andrew S. Tanenbaum, *Computer Networks*, Third edition.

2) Behrouz A. Forouzan, *Data Communications and Networking*, Third edition.

3) Douglas E. Comer, *Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture* (4th Edition).

4) James F. Kurose, *Computer Networking*: A Top-Down Approach Featuring the Internet (3rd Edition).

5) Larry L. Peterson, *Computer Networks*: A Systems Approach, 3rd Edition (The Morgan Kaufmann Series in Networking).

6) W. Richard Stevens, *The Protocols (TCP/IP Illustrated, Volume 1)*.

7) William Stallings, *Data and Computer Communications*, Seventh Edition.